

Robust and consistent estimation of generators in credit risk

Greig Smith*

University of Edinburgh
Maxwell Institute for Mathematical Sciences
School of Mathematics
Edinburgh, EH9 3FD, UK

G.Smith-13@sms.ed.ac.uk

Gonalo dos Reis†

University of Edinburgh
School of Mathematics
Edinburgh, EH9 3FD, UK
and
Centro de Matemática e Aplicaões
(CMA), FCT, UNL, Portugal
G.dosReis@ed.ac.uk

01h38, 01/03/2017

Abstract

Bond rating Transition Probability Matrices (TPMs) are built over a one-year time-frame and for many practical purposes, like the assessment of risk in portfolios, one needs to compute the TPM for a smaller time interval. In the context of continuous time Markov chains (CTMC) several deterministic and statistical algorithms have been proposed to estimate the generator matrix. We focus on the Expectation-Maximization (EM) algorithm by [BS05] for a CTMC with an absorbing state for such estimation.

This work's contribution is fourfold. Firstly, we provide directly computable closed form expressions for quantities appearing in the EM algorithm. Previously, these quantities had to be estimated numerically and considerable computational speedups have been gained. Secondly, we prove convergence to a single set of parameters under reasonable conditions. Thirdly, we derive a closed-form expression for the error estimate in the EM algorithm allowing to approximate confidence intervals for the estimation. Finally, we provide a numerical benchmark of our results against other known algorithms, in particular, on several problems related to credit risk. The EM algorithm we propose, padded with the new formulas (and error criteria), is very competitive and outperforms other known algorithms in several metrics.

Keywords: Likelihood inference, Credit Risk, Transition Probability Matrices, EM algorithm, Markov Chain Monte Carlo

2010 AMS subject classifications: Primary: 62M05; Secondary: 60J22, 91G40

JEL subject classifications: C13, C63 and G32

*G. Smith was supported by The Maxwell Institute Graduate School in Analysis and its Applications, a Centre for Doctoral Training funded by the UK Engineering and Physical Sciences Research Council (grant [EP/L016508/01]), the Scottish Funding Council, Heriot-Watt University and the University of Edinburgh.

†G. dos Reis acknowledges support from the *Fundação para a Ciência e a Tecnologia* (Portuguese Foundation for Science and Technology) through the project [UID/MAT/00297/2013] (Centro de Matemática e Aplicaões CMA/FCT/UNL).

1 Introduction

Credit ratings play a key role not just in the calculation of a bank's capital charge (amount of capital a bank must hold) but also are typically a requirement for corporations wishing to issue bonds. There are different agencies which provide firms with a rating and the credit rating the agency gives a company determines in some respect the financial health of the company. Typically ratings are of the form *AAA*, *AA*, *A*, *BBB*, *BB*, *B*, *C*, *D* (although it varies between agencies) with, *AAA* the best (safest), *C* the worst (riskiest) and *D* to imply the firm has defaulted. It is also standard for banks to use their own internal ratings system (see [YWZC14]). For an overview of the 'science' involved in the rating procedure see [Can04].

The main object of interest in this work is the so-called annual *Transition Probability Matrix* (TPM), it is a stochastic matrix which shows the migration probabilities of different rated companies within a year. Rating agencies produce these every year.

It is possible that such matrices are not initially stochastic due to company mergers or rounding for example. However, they can be renormalized by methods as described in [KS01] and, as argued in [BDK⁺02], renormalizing non rated companies across all ratings is indeed the industry standard.

The main problem considered here is that a TPM encases transition probabilities over a 1-year time frame and often in practice one needs a 3 month or 10 day transition matrix for which probabilities of default are lower than those in the TPM. Therefore one wants to accurately estimate the sub-annual matrix given the annual matrix. In the latest Basel proposals, Basel 3 ([Sup13, p.3]) a large part of the risk charge will be measured using ES (expected shortfall), which (as shown in [CDS10]) is extremely sensitive to a small shifts in probabilities. Therefore, accurate and consistent estimation is essential in the calculation.

Credit rating models within the Markovian framework are handy both from a theoretical and numerical perspective. Evidence is given in [LS02] that such Markovian structure is not true in practice, nonetheless, within the Markovian structure, efficient implementation of apt Markovian credit risk models and related risk measuring estimations able to deal with massive portfolios is a challenging problem, see [BMS14, RT15]. There have been several models that produce non-Markovian effects such as mixing two generators [FS08] or considering hidden Markov models [Kor12], see also [LKN⁺11]. All non-Markovian models require in one way or another access to additional data for accurate calibration. This data is costly, needs to be updated over time and many companies opt to deal only with the TPMs. This work focuses on practitioners that do not have access to the data. The issue of rating momentum will be dealt with in forthcoming research.

The problem at hand.

We take the view of a financial agent who wishes to estimate probabilities of default or assess risk in his portfolio due to credit transitions but does not have access to (the expensive) individual credit rating transitions. The agent only has the annual TPM, say $\mathbf{P}(1)$, and uses a continuous time Markov chain (CTMC), say $(\hat{\mathbf{P}}(t))_{t \geq 0}$, with a finite state space to model the changes in rating over time. Under standard conditions the evolution of the CTMC can be written as $\hat{\mathbf{P}}(t) = e^{\mathbf{Q}t}$ where \mathbf{Q} is the generator matrix. The problem is then to estimate \mathbf{Q} given $\mathbf{P}(1)$. This estimation is non-trivial due to the so-called embeddability problem (not reviewed here). It is discussed in great detail by [IRW01] and, for more of the mathematics and many of the existing results on the embeddability problem, we point the reader to [Lin11].

Several approaches exist to tackle this estimation problem [KS01, IRW01, TÖ04, BS05, Ina06, BS09], either using deterministic algorithms (e.g. diagonal or weighted adjustment, Quasi-optimization of the generator) or statistical ones (Expectation-Maximization (EM), Markov chain Monte-Carlo (MCMC) ones), see Section 3. We focus on the Expectation-Maximization algorithm of [BS05] for CTMCs and allow for an absorbing states.

Contribution of this work.

1. We provide closed form expressions for the expectations appearing in the EM algorithm. These quantities were previously expressed as integrals and were numerically estimated. This has allowed for computational speed up and further analysis of related quantities.
2. We provide sufficient conditions to extend the convergence result of [BS05] to individual parameters rather than just convergence of likelihoods.
3. We derive closed form expressions for the entries of the Hessian of the likelihood function used in the EM algorithm. This eliminated several instability issues appearing in numerical implementations found in related literature. Moreover, the result provides a way to estimate the error of the estimation and assess the nature of the stationary point the algorithm has converged to.
4. We give a short overview of known methods and implement them with some modifications as to improve their performance. See Sections 3 & 4 for precise meanings: we apply the algorithms to certain credit risk problems and carry out a simulation study to check the impact in the computation of *risk charges* (IRC with VaR, IDR with VaR and IRC with ES). We distinguish portfolio types (mixed, investment or speculative); the impact of different types of generators (stable vs unstable); dependence on the sample size and general convergence. We compare probabilities of default as maps of time across different algorithms and find interesting results.

For the study carried out, the implemented EM algorithm is very competitive. It is slightly slower than the deterministic algorithms but much faster than the MCMC algorithms. It embeds statistical properties like robustness that the deterministic algorithms cannot capture.

Remark 1.1. *We focus purely on continuous over discrete time models. The reason for this being that, continuous algorithms yield robust estimators while the discrete ones do not, where robustness is understood in the following sense: from $\mathbf{P}(1)$ estimate $\mathbf{P}(0.5)$ and $\mathbf{P}(0.25)$. From $\mathbf{P}(0.5)$ estimate $\mathbf{P}(0.25)$ again. Continuous algorithms yield the same $\mathbf{P}(0.25)$, discrete algorithms (in general) will not.*

This work is organized as follows. In Section 2 we present the EM algorithm and we state our main theoretical findings. In Section 3 we briefly present other known algorithms and in Section 4 we present the benchmarking results.

2 The EM Algorithm

There exists extensive literature on the majority of the algorithms we present in this paper, therefore we only provide brief discussions and include references for additional information. Further, we will use the theory of Markov chains extensively. We do not provide details of the theory, however, interested readers can consult texts such as [Nor98].

Preliminaries and standing convention. Throughout this manuscript we consider companies defined on a finite state space $\{1, \dots, h\}$, where each state corresponds to a rating. We denote *AAA* as rating 1 and *D* (default) as rating h . We adopt the standard notation that \mathbf{P} is an h -by- h stochastic matrix, which will be the observed TPM (at, say, time $t = 1$) and \mathbf{Q} is an h -by- h generator matrix. We further denote by $P_{ij} := (\mathbf{P})_{ij}$, by $q_{ij} := (\mathbf{Q})_{ij}$ and the intensity of state i by $q_i = \sum_{j \neq i} q_{ij}$ where $i, j \in \{1, \dots, h\}$. A standard assumption used in credit risk modeling is that default is an absorbing state, hence $P_{hh} = 1$. We work with infinitesimal generators of the following class.

Definition 2.1 (Stable-Conservative infinitesimal Generator matrix of a CTMC). *We say a matrix \mathbf{Q} is a generator matrix if the following properties are satisfied for all $i, j \in \{1, \dots, h\}$: i) $0 \leq q_{ij} < \infty$ for $i \neq j$; ii) $q_{ii} \leq 0$; and iii) $\sum_{j=1}^h q_{ij} = 0 \forall i$.*

If matrix \mathbf{Q} satisfies the above properties, then for all $t \geq 0$ the matrix $\mathbf{P}(t) := e^{\mathbf{Q}t}$ is a stochastic matrix, where $e^{\mathbf{A}}$ is the matrix exponential of matrix \mathbf{A} [Nor98, p.63]. The goal of the algorithms presented is thus to calculate a generator matrix \mathbf{Q} such that $e^{\mathbf{Q}t}$ is the best fit to the observed TPM, where t denotes the length of time between the rating updates (typically one year).

Throughout let $(X(t))_{t \geq 0}$ denote a CTMC over the finite state space $\{1, \dots, h\}$ with a generator \mathbf{Q} of the above class. Associated to $X(t)$ is, for i, j in the state space, $K_{ij}(t)$ the number of jumps from i to j in the interval $[0, t]$ and by $S_i(t)$ the holding time of state i in the interval $[0, t]$.

Remark 2.2. *If the matrix \mathbf{P} is embeddable¹, the algorithms below are pointless and one can easily tackle the problem through eigenvalue decomposition etc. Or in the case where the exact timing of rating transitions are known one can use the standard maximum likelihood estimator as in [JLT97].*

In our examples the only data given is a set of yearly TPMs which in general are not embeddable and therefore these methods mentioned do not yield useful results.

2.1 The Algorithm

Many methods have been developed in statistics in order to calculate the maximum likelihood estimate, but many methods break in the presence of missing data. Mathematically, we are interested in some set \mathcal{X} , but we are only able to observe \mathcal{Y} , with the assumption there is a many-to-one mapping from \mathcal{X} to \mathcal{Y} . That is, \mathcal{X} is a much richer set than \mathcal{Y} . When dealing with such a case, the Expectation Maximization (EM) algorithm often offers a robust solution to the problem. [MK07] provide a complete overview of the algorithm.

The basis of algorithm is, we observe data y which is a realisation (element) of \mathcal{Y} . We know y has density function g (sometimes referred to as a sampling density) depending on parameters Ψ in some space Λ , but we want the density (likelihood) of $\mathcal{X}(y)$. Hence, postulate some family of densities f , dependent on Ψ , where f corresponds to the density of the complete data set $\mathcal{X}(y)$ (the set of points $x \in \mathcal{X}$ which are in the pre-image of $y \in \mathcal{Y}$). The relation between f and g is,

$$g(y; \Psi) = \int_{\mathcal{X}(y)} f(x; \Psi) dx.$$

The idea is, the EM algorithm maximizes g w.r.t. Ψ , but we force it to do so by using the density f . Further, define,

$$R(\Psi'; \Psi) := \mathbb{E}_{\Psi} \left[\ln(f(x; \Psi')) | y \right] \quad \text{for } \Psi', \Psi \in \Lambda, \quad (2.1)$$

where $\mathbb{E}_{\Psi}[\cdot | y]$ is the conditional expectation, conditional on y under parameters Ψ . We assume R to exist for all pairs (Ψ', Ψ) , in particular we assume $f(x; \Psi) > 0$ almost everywhere in \mathcal{X} for all Ψ (otherwise the logarithm is infinite). Let us clarify, f is calculated using Ψ' , but the expectation is calculated using Ψ . The EM algorithm is then the following iterative procedure.

1. Choose an initial $\Psi^{(1)}$ and take $p = 1$.
2. E-step: Compute $R(\Psi; \Psi^{(p)})$.
3. M-step: Choose $\Psi^{(p+1)}$ to be the value of $\Psi \in \Lambda$ that maximizes $R(\Psi; \Psi^{(p)})$.
4. Check if the predefined convergence criteria is met, if not, take $p = p + 1$ and return to (ii).

¹In this setting a stochastic matrix \mathbf{P} is embeddable if there exists a generator \mathbf{Q} such that $\mathbf{P} = e^{\mathbf{Q}}$.

2.1.1 The particular problem of generator estimation

For our problem the observed process is a discrete time Markov chain (DTMC), the unobserved process we wish to estimate is a continuous time Markov chain (CTMC). Therefore, the observed data is the discrete transitions and the parameters we wish to estimate are the entries in the generator. The likelihood of a continuous time fully observed Markov chain with generator \mathbf{Q} is given by the following expression (see [KS97, Chapter 3.4]),

$$L_t(\mathbf{Q}) = \exp \left\{ \sum_{i=1}^h \left[\sum_{j \neq i} \log(q_{ij}) K_{ij}(t) - S_i(t) \sum_{j \neq i} q_{ij} \right] \right\},$$

with K and S the same as before. Hence given two generators \mathbf{Q}' , \mathbf{Q} , the function R in (2.1) is,

$$R(\mathbf{Q}'; \mathbf{Q}) = \sum_{i=1}^h \left[\sum_{j \neq i} \log(q'_{ij}) \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y] - \mathbb{E}_{\mathbf{Q}}[S_i(t)|y] \sum_{j \neq i} q'_{ij} \right], \quad (2.2)$$

where y denotes the discrete time observations. Maximizing for q'_{ij} in $R(\mathbf{Q}'; \mathbf{Q})$ yields

$$q'_{ij} = \frac{\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y]}{\mathbb{E}_{\mathbf{Q}}[S_i(t)|y]}. \quad (2.3)$$

The difficult step is the calculation of the expectations $\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y]$ and $\mathbb{E}_{\mathbf{Q}}[S_i(t)|y]$. We follow an approach similar to [BS05] (see also [BMNS02]) but express the result in a framework more suited to the problem of generator estimation from TPMs, rather than the estimation from individual movements. Furthermore, the result derived in [BMNS02] is for irreducible Markov chains making it not applicable to our case, accounted for in Proposition 2.4.

Consider the following functions (see [BMNS02]), for $1 \leq i, j \leq h$

$$V_{ij}^*(\mathbf{c}, \mathbf{Z}; t) = \mathbb{E}_{\mathbf{Q}} \left[\exp \left\{ - \sum_{\mu=1}^h c_{\mu} S_{\mu}(t) \right\} \prod_{\mu, \nu=1}^h Z_{\mu\nu}^{K_{\mu\nu}(t)} \mathbb{1}_{\{X(t)=j\}} \middle| X(0) = i \right], \quad (2.4)$$

where we denote by $\mathbf{c} = (c_1, \dots, c_h) \in \mathbb{R}^h$ and $\mathbf{Z} \in \mathbb{R}^{h \times h}$ with $Z_{ii} = 1$ for $i \in \{1, \dots, h\}$.

We observe that V_{ij}^* is the Laplace-Stieltjes transform of the holding times S and the probability generating function of the jumps K , with initial and final states $X(0) = i$ and $X(t) = j$ respectively. Denoting by $V^*(\mathbf{c}, \mathbf{Z}; t)$ the h -by- h matrix of elements $V_{ij}^*(\mathbf{c}, \mathbf{Z}; t)$. This allows us to give the main theorem (similar version) in [BMNS02].

Theorem 2.3. For $t \geq 0$, the matrix $\mathbf{V}^*(\mathbf{c}, \mathbf{Z}; t)$ is given by,

$$\mathbf{V}^*(\mathbf{c}, \mathbf{Z}; t) = \exp\{[\mathbf{Q} \bullet \mathbf{Z} - \Delta(\mathbf{c})]t\},$$

where \bullet is the Schur (Hadamard) product² of matrices, \mathbf{Q} is the generator matrix from the CTMC and $\Delta(\mathbf{c})$ is the diagonal matrix with entries c_i at position ii for $i = 1, \dots, h$.

This allows us to produce closed form expressions for the expectation terms in (2.3).

²The Shur product of two $h \times h$ matrices A and B is the $h \times h$ matrix with elements $A_{ij}B_{ij}$.

Proposition 2.4. Let \mathbf{e}_i be the column vector of length h which is one at entry i and zero elsewhere, further let us define the $2h$ -by- $2h$ matrices $\mathbf{C}_\gamma^{(\alpha\beta)}$ and $\mathbf{C}_\phi^{(\alpha)}$ as,

$$\mathbf{C}_\gamma^{(\alpha\beta)} = \begin{bmatrix} \mathbf{Q} & q_{\alpha\beta} \mathbf{e}_\alpha \mathbf{e}_\beta^\top \\ 0 & \mathbf{Q} \end{bmatrix} \quad \text{and} \quad \mathbf{C}_\phi^{(\alpha)} = \begin{bmatrix} \mathbf{Q} & \mathbf{e}_\alpha \mathbf{e}_\alpha^\top \\ 0 & \mathbf{Q} \end{bmatrix} \quad \alpha, \beta \in \{1, \dots, h\}. \quad (2.5)$$

Consider a Markov chain X that we observe at n time points $0 \leq t_1 < t_2 < \dots < t_n$ and denote by y_s the state of the Markov chain at time t_s , i.e. $y_s := X(t_s)$. Then, the expected jumps and holding times over the observations are,

$$\begin{aligned} \mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y] &= \sum_{s=1}^{n-1} \frac{\left(\exp\{\mathbf{C}_\gamma^{(ij)}(t_{s+1} - t_s)\} \right)_{y_s, h+y_{s+1}}}{\left(\exp\{\mathbf{Q}(t_{s+1} - t_s)\} \right)_{y_s, y_{s+1}}}, \\ \mathbb{E}_{\mathbf{Q}}[S_i(t)|y] &= \sum_{s=1}^{n-1} \frac{\left(\exp\{\mathbf{C}_\phi^{(i)}(t_{s+1} - t_s)\} \right)_{y_s, h+y_{s+1}}}{\left(\exp\{\mathbf{Q}(t_{s+1} - t_s)\} \right)_{y_s, y_{s+1}}}. \end{aligned}$$

Proof. We use the fact that \mathbf{V}^* in (2.4) satisfies the following differential equation (see [BS05]),

$$\frac{d}{dt} V_{\mu\nu}^*(\mathbf{c}, \mathbf{Z}; t) = -V_{\mu\nu}^*(\mathbf{c}, \mathbf{Z}; t) c_\nu + \sum_{r=1}^h V_{\mu r}^*(\mathbf{c}, \mathbf{Z}; t) q_{r\nu} Z_{r\nu} \quad \mu, \nu \in \{1, \dots, h\}. \quad (2.6)$$

To simplify the presentation, define for any two states $\mu, \nu \in \{1, \dots, h\}$, satisfying the *positive probability condition* $\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = \mu) > 0$, for $t > 0$,

$$\bar{\xi}_{\mu\nu}^{ij}(t) := \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(t) = \nu, X(0) = \mu] \quad \text{and} \quad \bar{\zeta}_{\mu\nu}^i(t) := \mathbb{E}_{\mathbf{Q}}[S_i(t) | X(t) = \nu, X(0) = \mu].$$

Note that the *positive probability condition* allows for the Markov process to be reducible. It is a trivial calculation to show that,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(t)|y] = \sum_{s=1}^{n-1} \bar{\xi}_{y_s y_{s+1}}^{ij}(t_{s+1} - t_s) \quad \text{and} \quad \mathbb{E}_{\mathbf{Q}}[S_i(t)|y] = \sum_{s=1}^{n-1} \bar{\zeta}_{y_s y_{s+1}}^i(t_{s+1} - t_s). \quad (2.7)$$

Before continuing we study the related quantities,

$$\xi_{\mu\nu}^{ij}(t) = \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) \mathbb{1}_{\{X(t)=\nu\}} | X(0) = \mu] \quad \text{and} \quad \zeta_{\mu\nu}^i(t) = \mathbb{E}_{\mathbf{Q}}[S_i(t) \mathbb{1}_{\{X(t)=\nu\}} | X(0) = \mu].$$

Again we only consider indices μ, ν s.t. $\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = \mu) > 0$, for $t > 0$. From standard conditional probability, the relationship between these quantities is,

$$\bar{\xi}_{\mu\nu}^{ij}(t) = \frac{\xi_{\mu\nu}^{ij}(t)}{\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = \mu)} \quad \text{and} \quad \bar{\zeta}_{\mu\nu}^i(t) = \frac{\zeta_{\mu\nu}^i(t)}{\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = \mu)}. \quad (2.8)$$

From (2.4) we see that

$$\left. \frac{\partial}{\partial Z_{ij}} V_{\mu\nu}^*(0, \mathbf{Z}; t) \right|_{\mathbf{Z}=\mathbf{1}} = \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) \mathbb{1}_{\{X(t)=\nu\}} | X(0) = \mu] = \xi_{\mu\nu}^{ij}(t),$$

and further,

$$\left. \frac{\partial}{\partial c_i} V_{\mu\nu}^*(\mathbf{c}, \mathbf{1}; t) \right|_{\mathbf{c}=\mathbf{0}} = \mathbb{E}_{\mathbf{Q}}[-S_i(t) \mathbb{1}_{\{X(t)=\nu\}} | X(0) = \mu] = -\zeta_{\mu\nu}^i(t),$$

where we denote by $\mathbf{Z} = \mathbf{1}$ and $\mathbf{c} = \mathbf{0}$, $Z_{ab} = 1$ and $c_a = 0$ for all $a, b \in \{1, \dots, h\}$. Applying the same operations to (2.6), we obtain the following differential equation,

$$\frac{d}{dt}\xi_{\mu\nu}^{ij}(t) = (\exp\{\mathbf{Q}t\})_{\mu i} q_{i\nu} \delta_{\nu j} + \sum_{r=1}^h [\xi_{\mu\nu}^{ij}(t) q_{r\nu}] , \quad \xi_{\mu\nu}^{ij}(0) = 0$$

with δ the standard Kronecker delta. By defining $\xi^{ij}(t)$ as the h -by- h matrix with μ, ν entry $\xi_{\mu\nu}^{ij}(t)$, where we define $\xi_{\mu\nu}^{ij}(t) = 0$ for μ, ν such that $\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = \mu) = 0$. From the previous expression we obtain,

$$\frac{d}{dt}\xi^{ij}(t) = q_{ij} \exp\{\mathbf{Q}t\} \mathbf{e}_i \mathbf{e}_j^\top + \xi^{ij}(t) \mathbf{Q}.$$

By considering a system of inhomogeneous differential equations the solution is given by,

$$\xi^{ij}(t) = \int_0^t \exp\{\mathbf{Q}s\} q_{ij} \mathbf{e}_i \mathbf{e}_j^\top \exp\{(t-s)\mathbf{Q}\} ds. \quad (2.9)$$

Through a similar argument we obtain ζ^i as,

$$\zeta^i(t) = \int_0^t \exp\{\mathbf{Q}s\} \mathbf{e}_i \mathbf{e}_i^\top \exp\{(t-s)\mathbf{Q}\} ds. \quad (2.10)$$

We solve this integral using the method in [VL78] as follows, consider the upper triangular matrix,

$$\mathbf{C} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ 0 & \mathbf{A}_2 \end{bmatrix},$$

where $\mathbf{A}_1, \mathbf{A}_2$ and \mathbf{B}_1 are h -by- h matrices. Noting that any integer power of an upper triangular matrix is also upper triangular. Then the exponential of \mathbf{C} is an upper triangular matrix. Hence,

$$\exp\{\mathbf{C}t\} = \begin{bmatrix} \mathbf{F}_1(t) & \mathbf{G}_1(t) \\ 0 & \mathbf{F}_2(t) \end{bmatrix}.$$

Recalling for matrix exponentials, $\frac{d}{dt}e^{\mathbf{C}t} = \mathbf{C}e^{\mathbf{C}t} = e^{\mathbf{C}t}\mathbf{C}$. Hence we can define,

$$\frac{d}{dt}e^{\mathbf{C}t} = \begin{bmatrix} \mathbf{F}_1(t) & \mathbf{G}_1(t) \\ 0 & \mathbf{F}_2(t) \end{bmatrix} \begin{bmatrix} \mathbf{A}_1 & \mathbf{B}_1 \\ 0 & \mathbf{A}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1(t)\mathbf{A}_1 & \mathbf{F}_1(t)\mathbf{B}_1 + \mathbf{G}_1(t)\mathbf{A}_2 \\ 0 & \mathbf{F}_2(t)\mathbf{A}_2 \end{bmatrix}.$$

Alternatively, we can write, $\dot{\mathbf{F}}_j(t) = \mathbf{F}_j(t)\mathbf{A}_j \implies \mathbf{F}_j(t) = \exp\{\mathbf{A}_j t\}$, for $j = 1, 2$ and $\dot{\mathbf{G}}_1(t) = \mathbf{F}_1(t)\mathbf{B}_1 + \mathbf{G}_1(t)\mathbf{A}_2$. Again using the same result for inhomogeneous differential equations we may write the solution to the differential equation $\dot{\mathbf{G}}_1(t) = \mathbf{F}_1(t)\mathbf{B}_1 + \mathbf{G}_1(t)\mathbf{A}_2$ as,

$$\mathbf{G}_1(t) = \int_0^t \mathbf{F}_1(s)\mathbf{B}_1 e^{-s\mathbf{A}_2} ds e^{t\mathbf{A}_2} = \int_0^t e^{s\mathbf{A}_1}\mathbf{B}_1 e^{(t-s)\mathbf{A}_2} ds.$$

Setting $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{Q}$ and $\mathbf{B}_1 = q_{ij} \mathbf{e}_i \mathbf{e}_j^\top$, this integral is precisely (2.9). Similarly, setting $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{Q}$ and $\mathbf{B}_1 = \mathbf{e}_i \mathbf{e}_i^\top$, this is (2.10). Using (2.5), (2.8) and (2.7) we obtain the required results. \square

Thus we obtain closed form expressions for these two key quantities. This approach differs from [BS05] where a numerical scheme (uniformization method) is used. This yields the relation we desire, however, in our example we have an observed TPM (or sequence of TPMs), \mathbf{P} , in the case of

equal observation windows, t in the interval $[0, T]$ (although it is trivial to generalize) the expectation can be expressed as,

$$\begin{aligned}\mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}] &= \sum_{u=1}^N \sum_{s=1}^h \sum_{r=1}^h P_{sr}^u(t) \frac{\left(\exp\{\mathbf{C}_{\gamma}^{(ij)} t\}\right)_{s,h+r}}{(\exp\{\mathbf{Q}t\})_{s,r}}, \\ \mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}] &= \sum_{u=1}^N \sum_{s=1}^h \sum_{r=1}^h P_{sr}^u(t) \frac{\left(\exp\{\mathbf{C}_{\phi}^{(i)} t\}\right)_{s,h+r}}{(\exp\{\mathbf{Q}t\})_{s,r}},\end{aligned}\tag{2.11}$$

where $N = T/t$ (the number of observations) and \mathbf{P}^u is the TPM of the u -th observation.

Remark 2.5 (The reducible case). *Previously, we only had observed transitions, hence they must have a non-zero probability of occurring. Here we can sum s and r over the full range because $\mathbf{P}_{sr}(t)$ acts as an indicator of possible transitions, that is, if $P_{sr}(t) = 0$ we set the s, r component as 0. Clearly, if $P_{sr}(t) > 0$, but $(\exp\{\mathbf{Q}t\})_{sr} = 0$, \mathbf{Q} is misspecified.*

Likelihood Convergence of the EM algorithm

In the case of this problem [BS05] provide a proof that the likelihood function converges with one small caveat in order to keep the parameter space compact. Namely, they use the following constrained parameter space, \mathcal{Q}_{ϵ} , which can be achieved by setting, $\mathcal{Q}_{\epsilon} = \{\mathbf{Q} \in \mathcal{Q} | \det[\exp(\mathbf{Q})] \geq \epsilon\}$ (\mathcal{Q} is the parameter space from Definition 2.1) for some $\epsilon > 0$. Theorem 4 in [BS05] states that the algorithm will converge to a stationary point of the likelihood or hit the boundary of the parameter space they have induced. It is accepted this is a crude approach to solving the problem and further analysis is needed when $\det[\exp(\mathbf{Q})] = \epsilon$. An alternative approach would be to use a penalized likelihood as discussed in [MK07, p.214].

Parameter convergence criteria

The above convergence is sufficient for one to conclude convergence of the likelihood. However, it is not sufficient for convergence of the parameters as one cannot state that the series of iterates $\mathbf{Q}^{(k)}$ converge ($\|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\| \rightarrow 0$ as $k \rightarrow \infty$). From a theoretical standpoint this may not be as important as convergence of the likelihood itself, it is of key importance in applications. For instance, without convergence of the parameters the risk charge different financial agents obtain from the same data may vary wildly, even under very strict convergence conditions. Before proving convergence we have two important points that we will require.

Remark 2.6. *With (2.11) in mind we assume that for any $s \neq r$ such that $P_{sr}^u(t) = 0$ for all u , we take the starting point $q_{sr}^{(0)} := (\mathbf{Q}^{(0)})_{sr} = 0$. As discussed in [BS05], any point set to zero will stay at zero for all iterations. Note, we are not changing the problem since these terms will converge to zero under the EM algorithm.*

Assumption 2.7 (Element constraint). *Similar to [BS05], we will use a manual space constraint to obtain the convergence. Take $1 > \epsilon > 0$, such that $\forall i \neq j, q_{ij} < 1/\epsilon$. Moreover, we assume adjacent mixing, namely, for $i \in \{2, \dots, h-1\}$, $q_{i,i\pm 1} > \epsilon$ and $q_{1,2} > \epsilon$.*

We denote the space of generator matrices which satisfy this condition as Λ_{ϵ} .

The above assumption ensures non-zero entries in the tri-diagonal band and also only finite entries as one can take ϵ as small as we wish. In the case of TPMs associated to credit ratings, such

an assumption is typically satisfied as one generally has diagonally dominant matrices and companies can always be upgraded or downgraded by one, thus $P_{i,i\pm 1}^u$ are typically non-zero. Diagonal dominance is sufficient for the generator to be identifiable and therefore entries do not blow up, we discuss the notion of identifiability in Section 2.2.

Proving the parameters converge is more challenging than the likelihoods, however, [Wu83] provide a sufficient condition for this to occur, namely a sufficient condition for $\|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\| \rightarrow 0$ as $k \rightarrow \infty$ is, there exists a forcing function³ σ such that,

$$R(\mathbf{Q}^{(k+1)}; \mathbf{Q}^{(k)}) - R(\mathbf{Q}^{(k)}; \mathbf{Q}^{(k)}) \geq \sigma(\|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\|).$$

An example of a forcing function is $\sigma(t) = \lambda t^2$ where $\lambda > 0$. We require the following bounds on the expected values to show convergence.

Lemma 2.8. *Let N and \mathbf{P}^u be as defined in (2.11) and assume for $i \neq j$ there exists a $u \in \{1, \dots, N\}$ such that $P_{ij}^u > 0$ (we observe a movement from i to j in observation window u). Then we obtain the following bounds on the expected number of jumps:*

$$P_{ij}^u \frac{\epsilon q_{ij}}{h} \leq \mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}] \leq h^2 N \frac{ht}{\epsilon \min\{\epsilon^h t^h \exp\{-th^2/\epsilon\}, \exp\{ht/\epsilon\}\}}. \quad (2.12)$$

Moreover, assuming there exists a $u \in \{1, \dots, N\}$ such that $P_{ii}^u > 0$, we obtain the following bound on the expected holding time,

$$\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}] \geq P_{ii}^u t \exp\{-ht/\epsilon\}. \quad (2.13)$$

In order to maintain the flow of the text we state our main convergence result, and the proof of the Lemma is deferred to Appendix A.1.

Theorem 2.9. *Under Assumption 2.7, then, there exists a $\lambda > 0$ such that for all EM iterations $k \in \mathbb{N}$,*

$$R(\mathbf{Q}^{(k+1)}; \mathbf{Q}^{(k)}) - R(\mathbf{Q}^{(k)}; \mathbf{Q}^{(k)}) \geq \lambda \|\mathbf{Q}^{(k+1)} - \mathbf{Q}^{(k)}\|^2,$$

where $\|\cdot\|$ is the Euclidean norm.

Proof. Writing out the difference in the R terms we obtain,

$$\sum_{i=1}^h \sum_{j \neq i} \left[\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(t)|\mathbf{P}] (\log(q_{ij}^{(k+1)}) - \log(q_{ij}^{(k)})) - \mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}] (q_{ij}^{(k+1)} - q_{ij}^{(k)}) \right].$$

Due to the form of the Euclidean norm squared and the function R , it is sufficient to show the inequality holds for all $i \neq j$. Namely, it is sufficient to show the existence of a $\lambda > 0$ such that,

$$\begin{aligned} \mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}] (\log(q_{ij}^{(k+1)}) - \log(q_{ij}^{(k)})) \\ - \mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}] (q_{ij}^{(k+1)} - q_{ij}^{(k)}) \geq \lambda (q_{ij}^{(k+1)} - q_{ij}^{(k)})^2, \end{aligned} \quad (2.14)$$

for all $i \neq j$. We tackle the log terms first. It is well known that we can express any C^∞ function using Taylor expansion to a finite number of terms with some error (remainder) term. Moreover,

³A forcing function is defined as any function $\sigma : [0, \infty) \rightarrow [0, \infty)$ such that for any sequence t_k defined in $[0, \infty)$, $\lim_{k \rightarrow \infty} \sigma(t_k) = 0$ implies $\lim_{k \rightarrow \infty} t_k = 0$.

the error term has a known form and hence, using so-called exact Taylor expansion to second order, there exists a $Z \in [\min(q_{ij}^{(k)}, q_{ij}^{(k+1)}), \max(q_{ij}^{(k)}, q_{ij}^{(k+1)})]$ such that,

$$\log(q_{ij}^{(k+1)}) - \log(q_{ij}^{(k)}) = \frac{-1}{q_{ij}^{(k+1)}}(q_{ij}^{(k)} - q_{ij}^{(k+1)}) + \frac{1}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2,$$

where we have expanded $q_{ij}^{(k)}$ around $q_{ij}^{(k+1)}$. Substituting (2.3) into the LHS of (2.14), the condition simplifies to,

$$\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \geq \lambda(q_{ij}^{(k+1)} - q_{ij}^{(k)})^2.$$

In order to show this bound we need to get a handle on Z . Clearly, there are two options between iterations, either $q_{ij}^{(k)} > q_{ij}^{(k+1)}$ or $q_{ij}^{(k)} \leq q_{ij}^{(k+1)}$. In the latter case we obtain,

$$\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \geq \frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}]^2}{2\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2.$$

Since we can element wise bound $\mathbf{Q}^{(k)}$, using Lemma 2.8 and Assumption 2.7 we can bound the term $\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]$ from above and $\mathbb{E}_{\mathbf{Q}^{(k)}}[S_i(T)|\mathbf{P}]$ from below by constants (depending on ϵ). Hence, we can choose a λ independent of k such that the condition is satisfied.

The second case $q_{ij}^{(k)} > q_{ij}^{(k+1)}$, follows a similar argument. Again, we can set Z as the larger of the two values, thus we obtain the following inequality,

$$\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \geq \frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2(q_{ij}^{(k)})^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2.$$

Using Lemma 2.8, we can reduce this inequality to,

$$\frac{\mathbb{E}_{\mathbf{Q}^{(k)}}[K_{ij}(T)|\mathbf{P}]}{2Z^2}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2 \geq \frac{P_{ij}^u \epsilon}{2h q_{ij}^{(k)}}(q_{ij}^{(k)} - q_{ij}^{(k+1)})^2.$$

Since $P_{ij}^u > 0$ and we can bound each q_{ij} from above, again we choose a λ independent of k and the result follows. \square

Starting value for the EM algorithm

The final point to discuss, is the choice of the initial matrix \mathbf{Q} . It is useful from a computational point of view to start in a good place. Here we choose \mathbf{Q} based on a generalization of the QOG algorithm (described in Section 3.1) that allows for complex inputs. For each entry q_{ij} we define the input as,

$$q_{ij} \rightarrow \text{sign}(\text{Re}(q_{ij})) \times |q_{ij}|,$$

where $|q_{ij}|$ is the magnitude of q_{ij} and $\text{Re}(q_{ij})$, is the real component of q_{ij} . With the newly defined \mathbf{Q} we apply the QOG algorithm. We take any zero entries not in the final row to be a small number (10^{-3} , say) unless there are zero observed transitions. This defines our initial choice of \mathbf{Q} . We define the EM algorithm steps as,

1. Take an initial intensity matrix \mathbf{Q} and positive value ϵ .

2. While the convergence criteria is not met and all entries of \mathbf{Q} are within the boundaries

- (1) E-step: calculate $\mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}]$ and $\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}]$.
- (2) M-step: set $q'_{ij} = \mathbb{E}_{\mathbf{Q}}[K_{ij}(T)|\mathbf{P}]/\mathbb{E}_{\mathbf{Q}}[S_i(T)|\mathbf{P}]$, for all $i \neq j$ and set q_{ii} appropriately.
- (3) Set $\mathbf{Q} = \mathbf{Q}'$ (where \mathbf{Q}' is the matrix of q' 's) and return to E-step.

3. End while and return \mathbf{Q} .

This leads to the following theorem for convergence of the EM.

Theorem 2.10 (Convergence of the EM). *Assume that our initial point is in the parameter space Λ_ϵ : is a true generator and satisfies Assumption 2.7. Then either the sequence of points $\{\mathbf{Q}^{(k)}\}_k$ converges to a single point in Λ_ϵ , or the entries go to the boundary (blow up or some tri-diagonal elements in the non-absorbing row go to zero).*

A proof of Theorem 2.10 follows directly from Theorem 4 in [BS05] and our Theorem 2.9.

2.2 Variance Estimation

In this section we derive an expression for the Hessian of the likelihood. We use a result in [Oak99] and follow [BS09], however, unlike [BS09], we provide a closed form expression for the Hessian. This result eliminates the stability problems observed in the numerical simulation case when the entries in \mathbf{Q} are small. The Hessian provides a way to estimate the standard error of the maximum likelihood estimates and further allows us to assess the nature of the converged stationary point (this is further discussed in Section 4.4).

We point the reader to [BS05, Theorem 1] for results on the existence and uniqueness of maximum likelihood estimators with respect to this problem. Further, for discussions on consistency and asymptotic normality related to this problem one should consult [KW13], [KW14]. [KW13], provide sufficient conditions for consistency, the key assumption relies on so-called model *identifiability*⁴. [KW13] prove *identifiability* under conditions which are too restrictive for our purpose; [BS05, DY07] discusses the problem of *identifiability* in detail. From [Cut73], [BS05] for the model to be identifiable it is sufficient to have $\min_i(\exp\{\mathbf{Q}t\})_{ii} > \exp\{-\pi\}$. Therefore, any diagonally dominant matrix (such as those in credit ratings) will have an identifiable generator (provided $e^{\mathbf{Q}t}$ is itself diagonally dominant). The crucial assumption in [KW14] to obtain asymptotic normality, is that the Hessian must be invertible at the true value, we can of course verify invertibility a posteriori.

We recall a result from [Oak99] for calculating the Hessian of the likelihood.

Lemma 2.11. *The second derivative of the likelihood with parameter Ψ and observed information y is related to the EM function R by*

$$\frac{\partial^2 L(\Psi; y)}{\partial \Psi^2} = \left[\frac{\partial^2 R(\Psi'; \Psi)}{\partial \Psi'^2} + \frac{\partial^2 R(\Psi'; \Psi)}{\partial \Psi' \partial \Psi} \right]_{\Psi' = \Psi}.$$

Injecting (2.2) in the above we obtain,

$$\frac{\partial^2 R(\mathbf{Q}'; \mathbf{Q})}{\partial q'_{\alpha\beta} \partial q'_{\mu\nu}} = \frac{-1}{q_{\mu\nu}^2} \mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y] \delta_{\alpha\mu} \delta_{\beta\nu}, \quad (2.15)$$

$$\frac{\partial^2 R(\mathbf{Q}'; \mathbf{Q})}{\partial q_{\alpha\beta} \partial q'_{\mu\nu}} = \frac{1}{q'_{\mu\nu}} \frac{\partial}{\partial q_{\alpha\beta}} \mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y] - \frac{\partial}{\partial q_{\alpha\beta}} \mathbb{E}_{\mathbf{Q}}[S_\mu(t)|y], \quad (2.16)$$

⁴In our setting a model is identifiable if there are no two generators $\mathbf{Q} \neq \mathbf{Q}'$ such that $\exp\{\mathbf{Q}t\} = \exp\{\mathbf{Q}'t\}$.

where δ_{ab} is the Kronecker delta. From our previous work, (2.15) is easy to obtain, however, (2.16) involves derivatives of the expected jumps and holding times and is thus challenging. [BS09] opt for a simple numerical scheme to compute these derivatives and found unstable results, although the authors do remark that more sophisticated numerical schemes could yield improved results at greater computational expense.

It is worth noting we have made no comment on the allowed values of α, β, μ and ν , other than the clear restriction that they belong to $\{1, \dots, h\}$. Let us now state the following definition.

Definition 2.12 (Allowed pairs). *We say that the pair α, β is allowed if $\alpha \neq \beta$ (not in the diagonal) and $q_{\alpha\beta}$ is not converging to zero under the EM algorithm.*

For practical applications, one can imagine the set of allowed values, as the set of α, β such that $q_{\alpha\beta} > \epsilon$, where ϵ is some cut-off point (10^{-8} , say). The reason we must exclude small parameters is, this analysis only holds in the large data limit, since we do not have an infinite amount of data we cannot for certain rule out some jump, however, if $q_{\alpha\beta}$ is converging to zero, it implies that this parameter is either zero or extremely close to zero and therefore we can bound it above by a small number. We now present the following theorem.

Theorem 2.13. *Let $\mu, \nu, \alpha, \beta \in \{1, \dots, h\}$, and \mathbf{Q}, \mathbf{Q}' be two generator matrices ($\in \Lambda_\epsilon$ for some $\epsilon > 0$). For any two allowed pairs α, β and μ, ν , the derivative in (2.16) is,*

$$\begin{aligned} \frac{\partial^2 R(\mathbf{Q}'; \mathbf{Q})}{\partial q_{\alpha\beta} \partial q'_{\mu\nu}} &= \sum_{s=1}^{n-1} \frac{1}{q'_{\mu\nu}} \left[- (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \right. \\ &\quad \left. + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(e^{\mathbf{C}_\psi^{(\alpha\beta, \mu\nu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \right] \\ &\quad - \left[- (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} (e^{\mathbf{C}_\phi^{(\mu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \right. \\ &\quad \left. + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(e^{\mathbf{C}_\omega^{(\alpha\beta, \mu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \right], \end{aligned}$$

where the $2h$ -by- $2h$ matrices, $\mathbf{C}_\gamma^{(\alpha\beta)}, \mathbf{C}_\phi^{(\alpha)}, \mathbf{C}_\eta^{(\alpha\beta)}$, are defined as,

$$\mathbf{C}_\gamma^{(\alpha\beta)} = \begin{bmatrix} \mathbf{Q} & q_{\alpha\beta} \mathbf{e}_\alpha \mathbf{e}_\beta^\top \\ 0 & \mathbf{Q} \end{bmatrix}, \quad \mathbf{C}_\phi^{(\alpha)} = \begin{bmatrix} \mathbf{Q} & \mathbf{e}_\alpha \mathbf{e}_\alpha^\top \\ 0 & \mathbf{Q} \end{bmatrix}, \quad \mathbf{C}_\eta^{(\alpha\beta)} = \begin{bmatrix} \mathbf{Q} & \mathbf{e}_\alpha \mathbf{e}_\beta^\top - \mathbf{e}_\alpha \mathbf{e}_\alpha^\top \\ 0 & \mathbf{Q} \end{bmatrix},$$

and the $4h$ -by- $4h$ matrices $\mathbf{C}_\psi^{(\alpha\beta, \mu\nu)}, \mathbf{C}_\omega^{(\alpha\beta, \mu)}$ are defined

$$\mathbf{C}_\psi^{(\alpha\beta, \mu\nu)} = \begin{bmatrix} \mathbf{C}_\gamma^{(\mu\nu)} & \frac{\partial \mathbf{C}_\gamma^{(\mu\nu)}}{\partial q_{\alpha\beta}} \\ 0 & \mathbf{C}_\gamma^{(\mu\nu)} \end{bmatrix}, \quad \mathbf{C}_\omega^{(\alpha\beta, \mu)} = \begin{bmatrix} \mathbf{C}_\phi^{(\mu)} & \frac{\partial \mathbf{C}_\phi^{(\mu)}}{\partial q_{\alpha\beta}} \\ 0 & \mathbf{C}_\phi^{(\mu)} \end{bmatrix}.$$

The proof of this uses similar techniques to Proposition 2.4, and is deferred to Appendix A.2.

Remark 2.14. *In the above representation for the derivative of R , we use subscripts of the form $h + y_{s+1}$ and $3h + y_{s+1}$, this is simply a consequence of the result in [VL78]. Namely, we are not interested in all the entries of the matrix, only an h -by- h segment. We therefore need to adjust the indexing to only take elements at this specific segment.*

Using Theorem 2.13 and Lemma 2.11, we can write the elements of the Hessian corresponding to the $q_{\alpha\beta}q_{\mu\nu}$ derivative as,

$$\begin{aligned} \frac{\partial^2 L(\mathbf{Q}; y)}{\partial q_{\alpha\beta} \partial q_{\mu\nu}} &= \sum_{s=1}^{n-1} \frac{-1}{q_{\mu\nu}^2} (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \delta_{\alpha\mu} \delta_{\beta\nu} \\ &\quad + \frac{1}{q_{\mu\nu}} \left[- (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \right. \\ &\quad \left. + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(e^{\mathbf{C}_\psi^{(\alpha\beta, \mu\nu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \right] \\ &\quad - \left[- (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} (e^{\mathbf{C}_\phi^{(\mu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \right. \\ &\quad \left. + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(e^{\mathbf{C}_\omega^{(\alpha\beta, \mu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}} \right]. \end{aligned}$$

A similar transform to (2.11) can be applied here to obtain the Hessian from TPMs. When using the result to estimate the error, some knowledge of the number of companies per rating is required.

2.2.1 Computation of the Error

Due to the Hessian only being defined for parameters $q_{\alpha\beta} > 0$, some parameters are not included in the Hessian, thus the matrix is smaller than h^2 -by- h^2 . We compute the Hessian as follows,

- Let N_a be the number of allowed points in the estimated \mathbf{Q} returned from the EM algorithm.
- Define an N_a -by-2 matrix $\mathbf{V}_\mathbf{Q}$ as the matrix which records the allowed (in the sense of Definition 2.12) components of \mathbf{Q} . The ij^{th} component of the Hessian is then the differential,

$$\frac{\partial^2}{\partial q_{\mathbf{V}_\mathbf{Q}(i,1)} \partial q_{\mathbf{V}_\mathbf{Q}(j,1)} \partial q_{\mathbf{V}_\mathbf{Q}(j,1)} \partial q_{\mathbf{V}_\mathbf{Q}(i,2)}}.$$

- If we denote the Hessian by the N_a -by- N_a matrix $\mathbf{H}(\cdot)$, then the information matrix is given by $-\mathbf{H}(\cdot)$. The estimated variance of the allowed parameter q_{ab} is then the i^{th} diagonal element of $-\mathbf{H}(\cdot)^{-1}$, where $\mathbf{V}_\mathbf{Q}(i, 1) = a$ and $\mathbf{V}_\mathbf{Q}(i, 2) = b$.
- Following standard statistics, the normal based 95% confidence interval for q_{ab} is $q_{ab} \pm 1.96 \sqrt{Var(q_{ab})}$.

3 Competitor Algorithms

3.1 Deterministic algorithms

Diagonal Adjustment (DA)

The first method to discuss is diagonal adjustment, see [IRW01]. Given a TPM, \mathbf{P} , one calculates the matrix logarithm directly. However, due to the embeddability problem, the logarithm may not be a valid generator. To solve this problem [IRW01] suggest setting for $i \neq j$,

$$q_{ij}^{DA} = \begin{cases} (\log(\mathbf{P}))_{ij}, & \text{if } (\log(\mathbf{P}))_{ij} \geq 0, \\ 0, & \text{otherwise.} \end{cases}$$

and adjusting (re-balancing) the diagonal element correspondingly, $q_{ii}^{DA} = \sum_{j \neq i} -q_{ij}$ for $i \in \{1, \dots, h\}$. Hence forcing the corresponding matrix \mathbf{Q}^{DA} to satisfy the properties of a generator.

Weighted Adjustment (WA)

Weighted adjustment is also suggested in [IRW01]. It follows diagonal adjustment except, one rebalances across the entire row. Again, calculate the logarithm of the TPM to find q 's, then compute

$$G_i = |q_{ii}| + \sum_{j \neq i} \max(q_{ij}, 0), \quad B_i = \sum_{j \neq i} \max(-q_{ij}, 0).$$

The entries corresponding to weighted adjustment are defined as,

$$q_{ij}^{WA} = \begin{cases} 0 & \text{if } i \neq j \text{ and } q_{ij} < 0, \\ q_{ij} - B_i |q_{ij}| / G_i & \text{otherwise if } G_i > 0, \\ q_{ij} & \text{otherwise if } G_i = 0. \end{cases}$$

Quasi-Optimisation of the Generator (QOG)

The above two methods are unfortunately not optimal in any sense. The QOG (Quasi-Optimisation of the Generator), method suggested in [KS01] relies on optimisation and is therefore an improvement on the diagonal and weighted adjustment methods. QOG involve solves the minimization problem $\min_{\mathbf{Q} \in \mathcal{Q}} \|\mathbf{Q} - \log(\mathbf{P})\|$, where \mathcal{Q} is the space of stable generator matrices and $\|\cdot\|$ is the Euclidean norm. Further, [KS01] provide an efficient algorithm to obtain \mathbf{Q} .

3.2 Statistical algorithm: Markov Chain Monte Carlo

An alternative statistical algorithm one can adopt is MCMC (Markov Chain Monte Carlo). For the reader's convenience we have included a summary of MCMC in Appendix B.

It should be noted that all MCMC algorithms presented here use a so-called auxiliary variable technique, by introducing the fully observed Markov chain, X as a random variable. Moreover, the prior for \mathbf{Q} is $\Gamma(\alpha, 1/\beta)$ (shape and scale), which is conjugate for the likelihood of a CTMC.

3.2.1 Gibbs Sampling - Bladt & Sorensen 2005

To simulate the Markov process, X , [BS05] suggest a rejection sampling method. As is stated in [BS05], such a sampling method runs into difficulties when considering low probability events since the rejection rate will be high (e.g. default for high rated bonds). The MCMC algorithm is summarised as follows, [Ina06],

1. Construct an initial generator \mathbf{Q} using the prior distribution $(\Gamma(\alpha_{ij}, 1/\beta_i))$.
2. For some specified number of runs
 - (1) Simulate X for each observation from Y , with law according to \mathbf{Q} .
 - (2) Calculate the quantities of interest K and S , from X .
 - (3) Construct a new \mathbf{Q} by drawing samples from $\Gamma(K_{ij}(t) + \alpha_{ij}, 1/(S_i(t) + \beta_i))$.
 - (4) Save this \mathbf{Q} and use it in the next simulation.
3. From the list of \mathbf{Q} s, drop some proportion (burn in), then take the mean of the remainder.

The issues with this method are the choice of α and β and the number of runs required before we know that the sample has converged (burn in). Both of these are critical in obtaining accurate answers from MCMC and although [BS05] suggested taking α_{ij} and β_i to be 1, they observe MCMC overestimating entries in the generator when true entries were small. Furthermore, here we are required to use the TPM indirectly through inferring company transitions. That is, we consider M companies in each rating and define the number of companies to make the transition i to j as $M \times P_{ij}$, this of course need not be an integer, but we can always normalise the entries. The reason we cannot use the TPM directly as we did in the EM algorithm is due to the fact that MCMC becomes very sensitive to the values in the prior. The burn in for MCMC will be of little concern to us here as will become apparent when carrying out analysis on the algorithms.

3.2.2 Importance Sampling - Bladt & Sorensen 2009

[BS09] address some of the issues contained in [BS05] by running the same algorithm as previous combined with an importance sampling scheme based on the Metropolis-Hastings algorithm (in its essence a single component Metropolis-Hastings algorithm). The proposal distribution suggested is a Markov chain with generator given by the ‘neutral matrix’ \mathbf{Q}^* , which takes the following form,

$$\mathbf{Q}^* = \frac{1}{W}(\mathbf{1}_h - \mathbf{I}_h - h\mathbf{I}_h),$$

where $\mathbf{1}_h$ and \mathbf{I}_h is the h -by- h matrix of ones and identity matrix respectively and W is a scaling factor set to match the intensities in the true generator matrix \mathbf{Q} . [BS09] note that if entries in \mathbf{Q} are known to be zero, then the corresponding element in \mathbf{Q}^* should also be set to zero and the diagonal modified accordingly. It is clear that here transitions rarely produced by the generated Markov chain will occur much more frequently under \mathbf{Q}^* . Thus we have solved (at least partially) one of the problems faced in MCMC. The importance sampling weights for a chain X are,

$$w(X) = \frac{L(\mathbf{Q}; X)}{L(\mathbf{Q}^*; X)},$$

where L is the CTMC likelihood. For the priors, [BS09] do not suggest any significant improvement on their earlier work. The authors use $\alpha = 1$ and $\beta = 5$, which they claim gives better results than the suggestion in [BS05]. However, it still provides a problem when dealing with entries in \mathbf{Q} which are close to zero. The problem stems from the fact that very little information is known (rarely observed) for certain transitions, therefore the output for these entries is mostly based on our prior beliefs.

3.2.3 MCMC Mode Algorithm

[Ina06] presented an alternative algorithm to the original MCMC algorithm presented in [BS05] whereby one calculates the mode rather than the mean. The author claims that this gives extremely accurate results and outperforms other algorithms. The reasoning presented is that the standard MCMC overestimates in the small probability cases due to the gamma distribution being ‘skewed’, therefore the mode is a better estimate. [Ina06] approximates the mode of $\{q_{ij}^{(k)}\}$ by kernel smoothing over the estimates (after taking the log transform to ensure all results are positive).

Remark 3.1 (Extension to MCMC). *There are many extensions and different MCMC methods that one could try to solve this problem (priors as hyperparameters or sequential Monte Carlo techniques for example). We consider here only those algorithms in the literature.*

4 Benchmarking the Algorithms

Due to the diversity of investments bank's make, a difficulty one faces when assessing these algorithms is that we cannot consider a single test. With this in mind we consider a host of tests on different portfolios and matrices. The computations were carried out on a Dell PowerEdge R920 with four Intel Xeon E7-4830 processors.

The first observation we make is, transitions matrices can vary substantially dependent on the financial climate (see [CHL04] and [Can04]). Therefore we consider two different generator matrices which can be thought of as the generator in financial stress and the generator in financial calm. In order to keep these matrices 'reasonable' we start off with the generator given in [CHL04] (built using a large amount of data) and consider a generator which has in general higher transition rates and one with lower transition rates. Through considering more than one generator we feel we provide a more detailed assessment of the performance of the various algorithms than other comparative reviews, such as [Ina06]. The generators we consider are shown in Table 4.1 and Table 4.2. We observe that Table 4.1, has more non-zero entries and larger entries than that of Table 4.2.

	Aaa	Aa	A	Baa	Ba	B	Caa	D
Aaa	-0.146371	0.085881	0.04549	0.015	0	0	0	0
Aa	0.018506	-0.166337	0.114831	0.033	0	0	0	0
A	0.0276	0.047012	-0.198043	0.09043	0.023001	0.01	0	0
Baa	0.011469	0.010734	0.088133	-0.243046	0.077569	0.044407	0.010734	0
Ba	0	0	0.019159	0.184699	-0.323077	0.106166	0.013053	0
B	0	0	0.012280	0.034822	0.093489	-0.296265	0.134273	0.022401
Caa	0	0	0	0	0.02	0.140209	-0.480939	0.320730
D	0	0	0	0	0	0	0	0

Table 4.1: True unstable generator

	Aaa	Aa	A	Baa	Ba	B	Caa	D
Aaa	-0.061371	0.055881	0.005490	0	0	0	0	0
Aa	0.013506	-0.096337	0.074831	0.008	0	0	0	0
A	0	0.037012	-0.107442	0.07043	0	0	0	0
Baa	0	0.000734	0.058133	-0.120843	0.057569	0.004407	0	0
Ba	0	0	0.009159	0.104699	-0.190024	0.076166	0	0
B	0	0	0	0.024822	0.083489	-0.174985	0.064273	0.002401
Caa	0	0	0	0	0	0.080209	-0.300939	0.220730
D	0	0	0	0	0	0	0	0

Table 4.2: True stable generator

Throughout the analysis we refer to the multiple MCMC algorithms introduced in Section 3 which we label in the following way: MCMC BS05 is [BS05]'s algorithm of Section 3.2.1; MCMC BS09 is [BS09]'s algorithm of Section 3.2.2; and MCMC Mode is [Ina06]'s algorithm in Section 3.2.3.

4.1 Sample Size Inference

The first test we consider is an extension to a test in [Ina06], where the author considers a true underlying generator and masks it by using it to simulate TPMs, which we view as observations, then applying the algorithms to each observation. The key point here is, [Ina06] only simulates 100 companies per rating and hence the outputted TPM is non-embeddable (has 0 entries for accessible jumps). This is an extremely useful test because it provides a fair and intuitive way to assess the performance of each algorithm, however, [Ina06] only considers one true generator and only one

level of information i.e. 100 companies per rating. Alongside the two different generators we also consider a range of companies per rating to determine its effect on convergence for each algorithm. Furthermore, [Ina06] uses seven years worth of data, although one would likely have access to multiple years worth of TPM data, it is unlikely that we would have seven years of transitions from the same generator. Hence we consider four years worth of data, which is more consistent with time homogeneity estimates for generators (see [CHL04]).

We calculate our estimates for the generator as follows.

1. Take a range of obligors per rating, here [50, 150, 300, 500, 750, 1500, 4000, 10 000] and also a set of 10 random seeds.
2. For each true generator simulate four one year TPMs for each seed and for each obligor per rating. Hence we have ($\# \text{Years} \times \# \text{Obligors categories} \times \# \text{Random Seeds} \times \# \text{True generators}$), simulated TPMs.
3. For each set of four simulated TPM we estimate the generator for each algorithm. MCMC may take a long time to run, therefore we consider the time taken to carry out the first 10 runs and the total time taken, if these exceed 80 or 1800 seconds respectively, the algorithm is deemed to be too slow and no result is returned. Note, MCMC algorithms use 1000 runs with a burn in of 100. This is smaller than [Ina06] for example, however, [Ina06] shows apparent convergence to the stationary distribution in a small number of iterations and we observe a similar result.
4. Therefore, for each algorithm we have ($\# \text{Obligors categories} \times \# \text{Random Seeds} \times \# \text{True generators}$) estimated generators to analyze.

We analyze the estimated generators by considering, distance between estimated generator and true generator in Euclidean norm and difference in one year probability of default. All results presented have been obtained by analyzing the estimated generator for each seed, then averaging. This gives a better picture of the average performance.

Algorithms	Deterministic	EM	MCMC
Time (seconds)	< 1	~ 10	~ 500

Table 4.3: Order of time taken to execute the various algorithms.

4.1.1 Convergence in Euclidean Norm

Our goal in this analysis is to consider the empirical rate of improvement of each algorithm as our ‘information’ about the true generator increases. For each obligor category we calculate the natural log of the distance (measured by the Euclidean norm) between the estimate and the true. The results are shown in figures 4.1 and 4.2.

Note the x -axis is on a logarithmic scale. We observe similarities between the two figures, most notably in the case of low information all algorithms have very similar convergence results, however, as we increase the information there is substantial variation in improvement, MCMC algorithms do not improve as well as the EM and deterministic algorithms. Missing points stem from an algorithm failing the acceptance times.

MCMC algorithms seem to hit a limit with improvement as information increases. This is possibly due to Monte Carlo error, but to improve this would require a large computational expense. For the [BS09] algorithm the neutral matrix approximation may give poor mixing, thus the additional error.

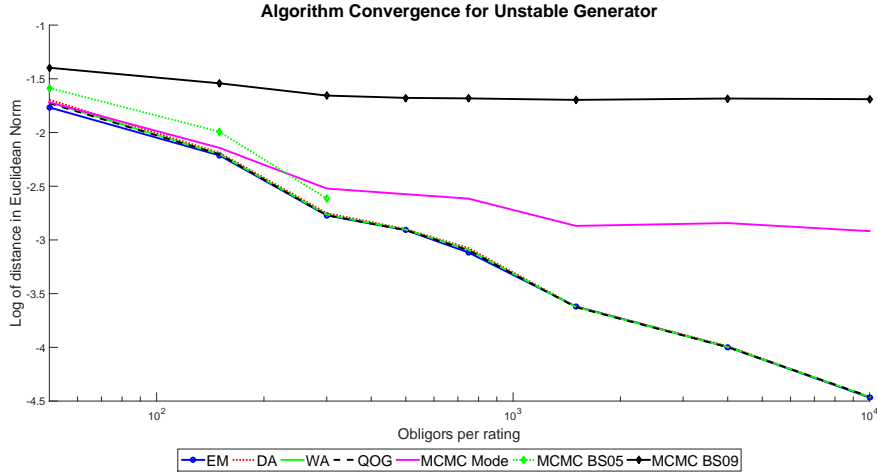


Figure 4.1: Showing the log of the error for each algorithm as a function of obligors per rating.

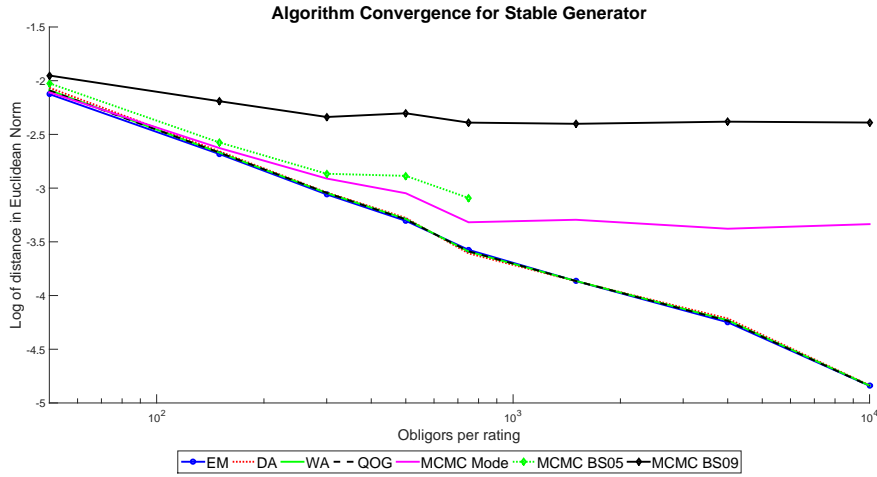


Figure 4.2: Showing the log of the error for each algorithm as a function of obligors per rating.

4.1.2 Error in Probability of Default

Although overall error is important, it does not provide details on the small probability scale. This is extremely important in banking, since estimation of the probability of default is crucial. Using the same estimated generators as previous we calculate the corresponding one year TPM, that is, we calculate $\exp\{Q_{\text{estimate}}\}$ (using the `expm` function in MATLAB) for each seed then take the average. The averaged TPM default probabilities are compared to the true ones. To keep the numbers in the comparisons meaningful we plot the log of the relative error, where we define,

$$\text{Relative Error} = \frac{|PD_{\text{estimate}} - PD_{\text{true}}|}{PD_{\text{true}}}.$$

The results of which are given in Figures 4.3 and 4.4.

Unlike the overall error, there appears to be far greater volatility in the error estimation w.r.t. the probability of default. Moreover, there appears to be no general downward trend in error for the investment grade ratings. A likely cause for this is, even with 10 000 companies there are still no/few investment grade defaults. MCMC performs worse than the other algorithms for the same reasons as previously discussed. The EM algorithm however, is consistently one of the smallest errors in all

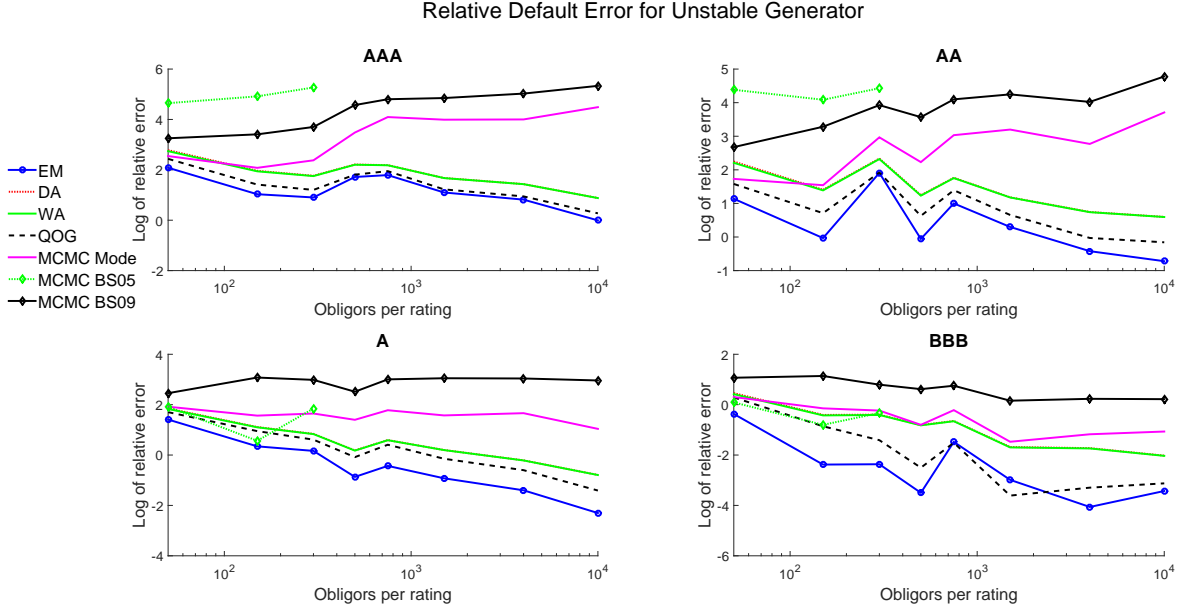


Figure 4.3: Showing the log of the relative default error for each algorithm as a function of obligors per rating.

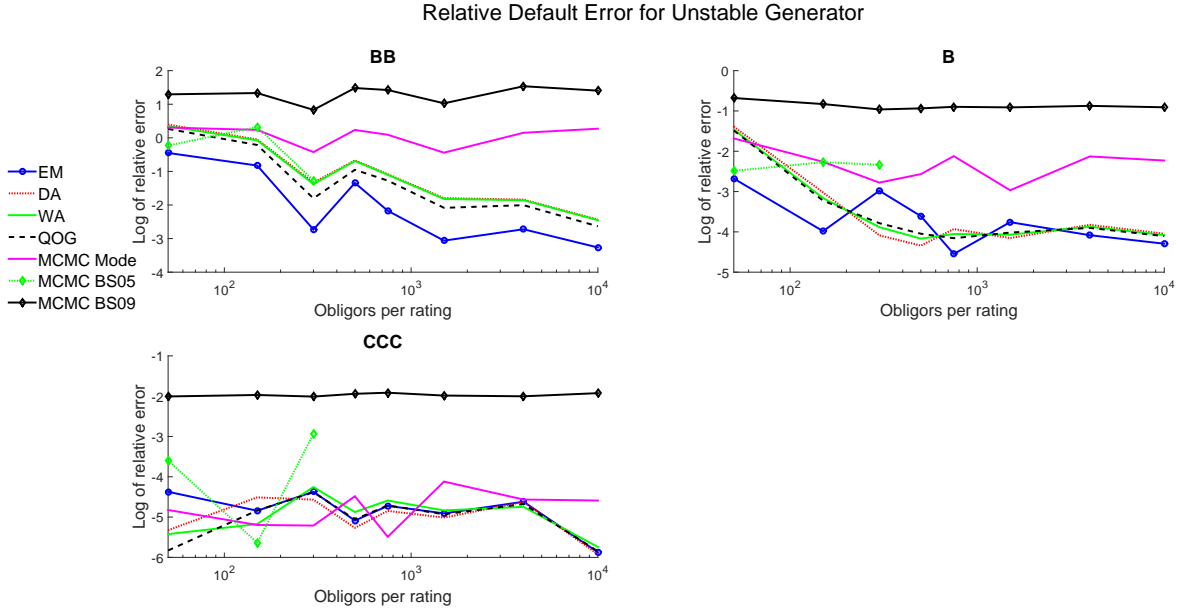


Figure 4.4: Showing the log of the relative default error for each algorithm as a function of obligors per rating.

ratings. We have only shown the results for the unstable generator, the stable generator followed a similar pattern.

4.2 Time Dependent Probability of Default

A key question that has not been addressed in the literature is how do the probabilities of default change in time among the several algorithms. As MCMC overestimates the small values of the gener-

ator, we only carry out this analysis with the EM, QOG and WA.

Similar to before, we start with a non-embeddable TPM, then estimate the generator matrix \mathbf{Q} , from \mathbf{Q} we can easily calculate the probability of a company with some initial rating defaulting in time $t > 0$, namely for any t we can calculate the full transition matrix as $e^{\mathbf{Q}t}$ and take the final column as the probability of default. The goal here is to assess how that probability changes with time. The TPM is given in Table 4.2. This TPM is similar to the TPMs one observes from the data, but also has various levels of ‘stability’ in different ratings and therefore we can assess how the algorithms treat the probability of default for varying probability.

	Aaa	Aa	A	Baa	Ba	B	C	D
Aaa	0.8824	0.1176	0	0	0	0	0	0
Aa	0.0064	0.9111	0.0813	0.0008	0.0001	0	0.0003	0
A	0.0003	0.0559	0.8836	0.0499	0.0079	0.0015	0.0002	0.0007
Baa	0	0.0116	0.1585	0.7640	0.0528	0.0070	0	0.0061
Ba	0	0	0.0213	0.1193	0.7746	0.0623	0.0099	0.0127
B	0	0	0.0062	0.0199	0.1669	0.7017	0.0730	0.0322
C	0	0	0	0	0.0417	0.2083	0.4544	0.2956
D	0	0	0	0	0	0	0	1

Table 4.4: Observed TPM used to estimate the generators in probability of default plots.

The probability of default across ratings over the one year time horizon is found in Figure 4.5.

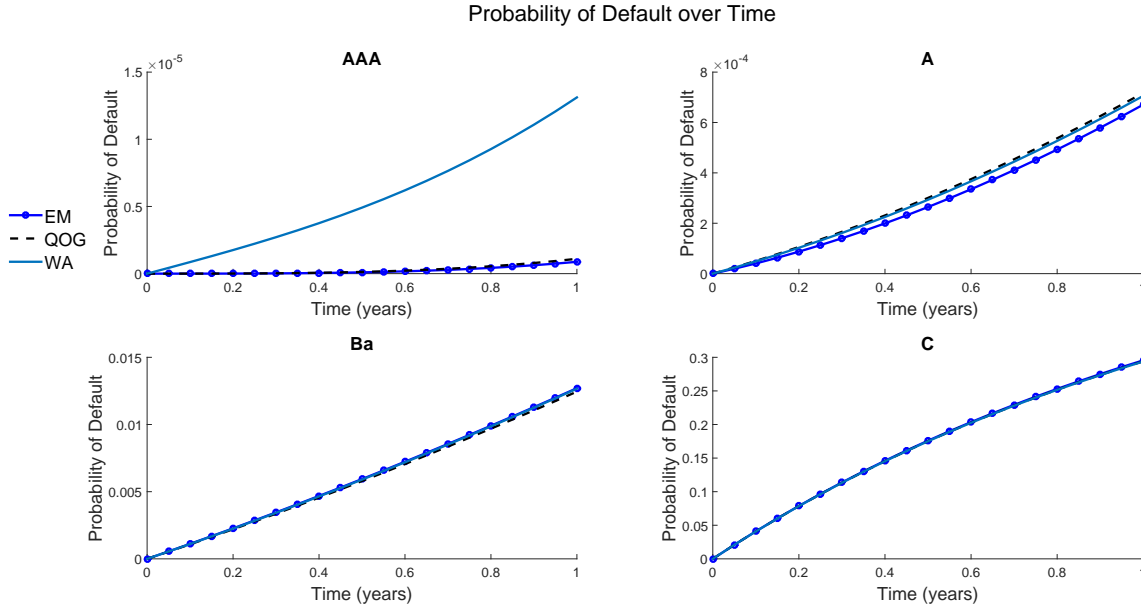


Figure 4.5: Probability of default over time for EM, QOG and WA.

The plots give a deeper understanding to the algorithms themselves. As the probability of default increases the algorithms converge, however, in the case of less defaults we observe a much larger discrepancy. This can be thought of as the algorithm’s ability to deal with missing data, in the lower grades we observe defaults and thus have a handle on the probability, however, in the case of AAA ratings we observe no defaults and therefore it is an approximation by the algorithm. This shows the difference between optimization based methods (QOG and EM) versus the more naive WA method.

4.3 Risk Charge

The previous tests have been rather theoretical, we now consider a practical test to assess the performance of these algorithms in calculating risk charges. Here we consider multiple portfolios to represent the risk appetites of different banks. The risk charges we consider are IRC (VaR at 99.9% with a 3 months liquidity horizon including mark to market loss), IDR (VaR at 99.9% over one year only considering default) and a theoretical risk charge which is IRC but measured using Expected Shortfall (ES) at 97.5%. The final risk charge is included due to the Basel committee showing an increasing interest in ES. We consider 4 years worth of simulated data, and to keep the analysis realistic we consider 300 companies per rating. Here we consider 3 different portfolios corresponding to risk adverse (all investment grade), a speculative portfolio (all speculative grades) and finally a mixed portfolio. The portfolios considered are given in Tables 4.5, 4.6 and 4.7. The tables show the values and ratings of the various bonds in each portfolio.

Aaa	100, 500, 1500, 750
Aa	200, 750, 2000, 650
A	150, 400, 400
Baa	300, 500, 150, 1500
Ba	500, 250, 700
B	200, 500
Caa	100, 150, 200

Table 4.5: Mixed portfolio

Aaa	1000, 500, 1500, 1500
Aa	100, 400, 750, 2000, 400, 1500
A	150, 100, 800, 400, 200
Baa	
Ba	
B	
Caa	

Table 4.6: Investment portfolio

Aaa	
Aa	
A	
Baa	
Ba	1000, 150, 100, 800, 1500
B	100, 300, 400, 750, 2000, 1500
Caa	400, 500, 400, 1000

Table 4.7: Speculative portfolio

Alongside these portfolios we calculate the risk charges using the following information,

- The interest rates we receive for a bond in each rating are

Aaa	Aa	A	Baa	Ba	B	Caa
2%	3%	4.2%	6.5%	10.2%	13.6%	25%

- We assume that all money is lost in the case of default (zero recovery rate).
- We calculate credit migration using the one factor⁵ credit metrics model ([GFB97]), i.e. normalised asset returns follow,

$$z_i = \beta_i X + \sqrt{1 - \beta_i^2} \epsilon_i,$$

where X is the systematic risk, ϵ_i is the idiosyncratic risk both standard normally distributed and β_i is the correlation to the systematic risk. Where β_i is as defined in [Sup03, p.50],

$$\beta_i = 0.12 \left(\frac{1 - \exp\{-50 P_i^D\}}{1 - \exp\{-50\}} \right) + 0.24 \left(1 - \frac{1 - \exp\{-50 P_i^D\}}{1 - \exp\{-50\}} \right),$$

where P_i^D is the probability of default of asset i . Consequently we see that the higher P_i^D the lower the value of β .

- Although more sophisticated methods are available for calculation of VaR and ES (see [Fer14]), we calculate the risk charges using Monte Carlo. This works here since the portfolios are small relative to a typical bank portfolio, therefore we can obtain accurate estimates using a reasonable number of simulations.

⁵This is technically not the true regulation for the calculation of IDR which requires a two factor model, however our goal here is only to use these calculations as a method for comparing algorithms.

- To make the results representative we simulate 300 obligors per rating for four years. Again, we calculate 10 realizations of the TPMs and estimate a generator for each realization.

We consider 15×10^5 simulations for each portfolio, to assess whether this was sufficient we calculated VaR and ES using 7.5×10^5 , 10×10^5 , 12.5×10^5 and 15×10^5 simulations and found the difference between 7.5×10^5 and 15×10^5 to be $< 5\%$ for all cases. Hence we are confident that 15×10^5 gives sufficiently accurate results for our purposes.

With respect to risk charge calculation, similar to the previous analysis, we calculate the risk charges for every set of TPMs, then average over all the seeds to obtain the risk charge. The risk charges as set by the true generators are given in Table 4.8.

	Stable			Unstable		
	Mixed	Investment	Speculative	Mixed	Investment	Speculative
IRC	£1104.70	£7.60	£4612.70	£706.30	£6.00	£3399.70
IRC ES	£507.70	£3.70	£2409.60	£749.50	£6.90	£3399.00
IDR	£750	£0	£3400	£1650	£150	£4300

Table 4.8: Risk charge results for the true generators.

To assess the performance of each algorithm we measure the error by the following,

$$\text{Risk Error} = \frac{\frac{1}{N} \sum_{i=1}^N |\text{Risk Charge Estimate}(i) - \text{Risk Charge True}|}{\text{Risk Charge True}},$$

where Risk Charge Estimate(i) is the i^{th} realization of the risk charge and N is the number of TPM sets (10 here). The results obtained by the algorithms are shown in Table 4.9.

		Stable			Unstable		
		Mixed	Investment	Speculative	Mixed	Investment	Speculative
IRC	EM	2.4	661	2.5	18.5	4644	4.6
	DA	5	662	1.9	46.2	6480	4.2
	WA	4.8	662	2.2	46.1	6424	4.2
	QOG	3.3	652	2.3	21.5	4851	4.2
	MCMCBS05	167	29150	3	94	26171	4.3
	MCMCBS09	15.8	1343	12.4	60.4	13395	9.5
	MCMCMode	8.6	1326	4.6	34.9	6618	3.9
IRC ES	EM	2.8	144	2.6	8.4	817	4.4
	DA	3.4	154	2.6	13.1	1236	3.9
	WA	3.1	155	2.5	13.1	1222	4
	QOG	2.8	121	2.5	6.9	841	4.1
	MCMCBS05	73.8	18243	4	53.4	14884	4.4
	MCMCBS09	14.2	600	12.2	40.2	5206	9.1
	MCMCMode	6.9	498	5.4	18.4	2387	3.8
IDR	EM	4	£40	3.6	5.4	280	3.3
	DA	5.3	£40	0.6	9.1	460	2.6
	WA	5.3	£40	0.6	9.4	460	2.8
	QOG	4.7	£40	1.2	4	390	2.7
	MCMCBS05	168	£1756	1.9	34.8	1233	1.7
	MCMCBS09	16	£80	7.8	27.9	1067	7
	MCMCMode	8.7	£80	3.3	12.4	613	2.3

Table 4.9: Risk charge results for each algorithm as a %.

It should be noted, in the stable IDR case all algorithms produce a non-zero value for the investment portfolio, therefore we have inserted the money value. The first observation we make is, all algorithms overestimate the risk for the investment portfolio. This is mainly due to the ‘step like’ nature of VaR, where in a small portfolio, small probability changes can make a large difference. In the other portfolios the algorithms (other than [BS05]) produce reasonable risk charges under both generators.

We conclude that the EM algorithm provides the best across the board results, performing best in the case of the mixed portfolios and is close to, if not the best in the other categories.

4.4 Error estimation of the EM algorithm

In a similar fashion to the analysis we have carried out previously we now test the error estimate given by the EM. Again, we mask the true generator by using simulated TPMs, however, here we only consider the scenario of 300 obligors per rating, but the number of years worth of data is varied. That is, we simulate 50 years worth of TPMs and then apply the EM algorithm using 1 years worth then 2 years etc up to 50 years. This analysis shows both the estimated error for the parameter and also how the error changes when more information is added. It should also be noted that we do not replace companies who have defaulted nor do we renormalise to have the same number of companies in each rating at the start of each year.

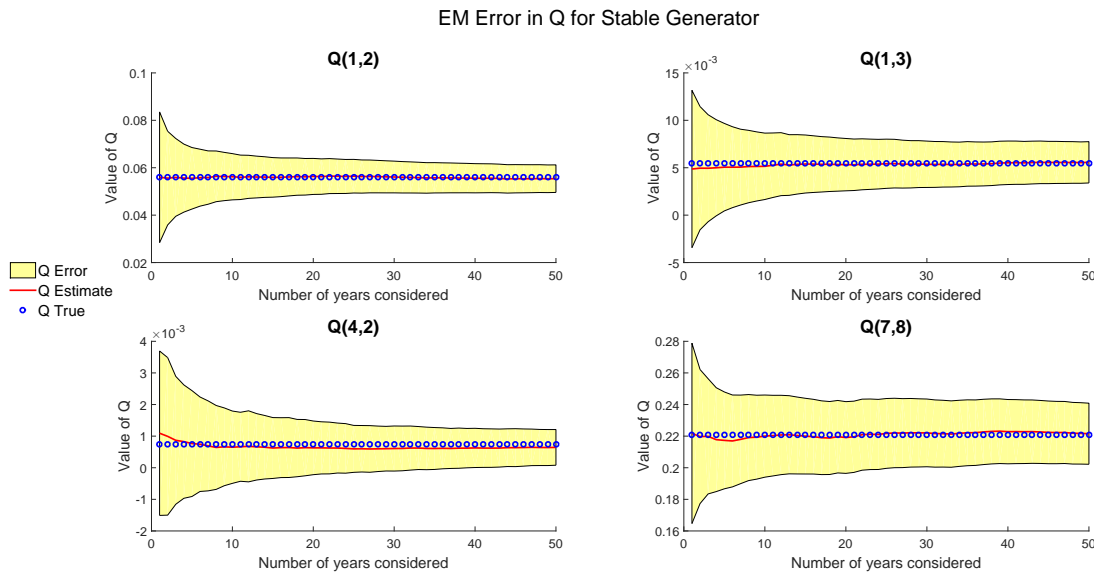


Figure 4.6: Showing the estimated 95% confidence interval for parameters as a function of years.

The transitions shown in Figures 4.6 and 4.7 were chosen to show a spectrum of the magnitudes in the generators, the other entries not shown are similar. The first point to make is that the true value of the parameter always lies within the confidence interval and the confidence interval shrinks as the number of years increases. Further, the estimate from the EM algorithm is extremely close to the true value in all cases, which backs up the claim we can be confident the EM converges to a point close to the maximum likelihood. The only graph that shows little improvement after the first few years is the $Q(7,8)$ transition in Figure 4.6, however, the EM estimated parameter appears to oscillate around the true, which hints that the likelihood is fairly constant around the true. The final point to make is, although some confidence intervals go below zero by a small amount, this is only true in the case where the parameter is extremely close to zero initially and further, once more data

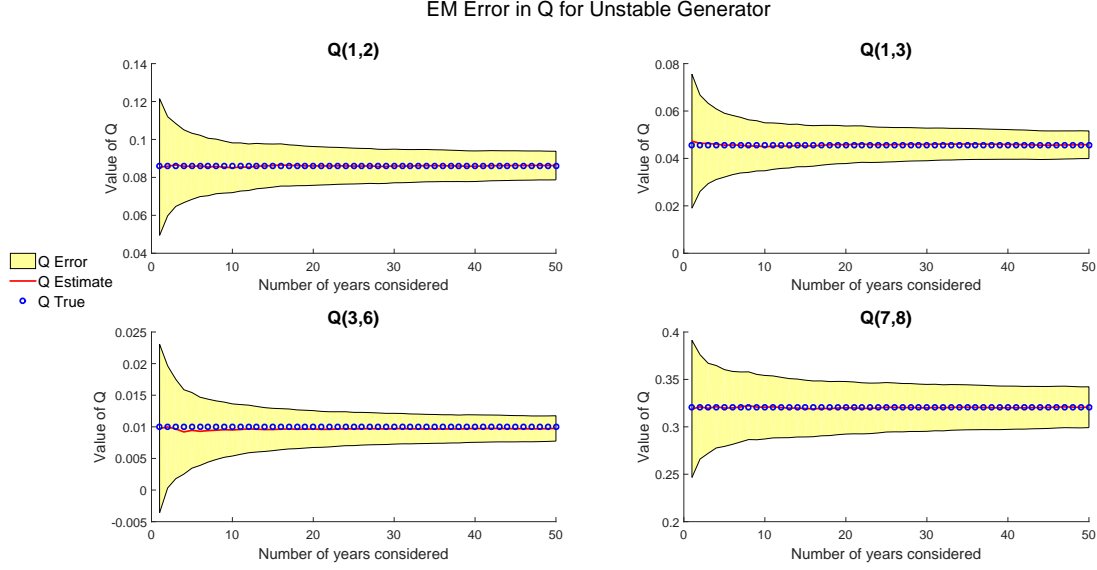


Figure 4.7: Showing the estimated 95% confidence interval for parameters as a function of years.

is considered, all parameters have confidence interval which are strictly positive.

Connection to the Global Maximum

A previous problem with the EM is we cannot be sure of the nature of the stationary point. However, we know the form of the Hessian, and therefore we can easily check if this point is a maximum by assessing the eigenvalues of this matrix. Clearly, if we were not at a maximum, then it would be worth perturbing the outputted generator and rerunning the algorithm.

Although it is useful to know whether or not the EM has converged to a maximum, one will naturally ask, is this maximum the global maximum? Computation of the global maximum in this problem has been a long standing issue which becomes highly non-trivial for dimensions above three, see [BS05] for further details.

Remark 4.1. *One way that has been suggested to improve the chances of the EM converging to the global maximum is, to start from multiple points. Here we can consider creating starting points by setting for each $i \neq j$, $q_{ij} \sim \text{Exp}(\lambda)$ for an appropriate λ then setting q_{ii} appropriately.*

We tested the EM according to the above remark and found in every case considered the EM always returns the same generator. Moreover, since we can evaluate derivatives of the likelihood analytically methods such as the Gradient descent could be used for further optimization.

5 Conclusions and future research

In this manuscript we deduced closed form expressions for the expected number of jumps and holding times of a CTMC with an absorbing state, over given observations and used the results to produce an optimized version of the EM algorithm to tackle the estimation of the generator matrix of the CTMC. The techniques used allowed us to derive a closed form expression for the Hessian of the likelihood function and use it to control the estimation error of the EM (with little computational expenditure).

Across the battery of tests carried out, the EM algorithm performs better than competing algorithms. The EM is a tractable algorithm which is slower than the deterministic algorithms but still

much faster (at least one order of magnitude) than the several Markov-Chain Monte-Carlo alternatives (Table 4.2). The statistical algorithms (EM and MCMC) embed a strong robustness property for the estimator contrary to the deterministic algorithms, i.e. the likelihood is far less sensitive to small changes in the underlying TPM. The EM, like the deterministic algorithms shows good convergence properties as the amount of data is increased. On the other hand, MCMC algorithms do not show the same trend, see Figs 4.1 and 4.2. In terms of estimation of risk charges, the EM algorithm is strongly competitive in all cases and outperforms in many of the scenarios.

On the more practical side, Figure 4.5 highlights that for lower ratings all algorithms produce essentially the same estimates for the probabilities of default while a palpable difference emerges at higher ratings. The deterministic algorithms seem to be overestimating the probability of defaults.

Lastly, non-Markovian phenomena like rating momentum (see [LS02]) and appropriate models to tackle it will be addressed in forthcoming research.

Acknowledgement(s)

The authors would like to thank Dr. R. P. Jena at Nomura Bank plc London for the helpful comments. In addition, the authors would like to thank Ruth King (U. of University), Ioannis Papastathopoulos (U. of University) and Samuel Cohen (Oxford Uni.) for the helpful discussions.

Funding

G. Smith was supported by The Maxwell Institute Graduate School in Analysis and its Applications, a Centre for Doctoral Training funded by the UK Engineering and Physical Sciences Research Council (grant [EP/L016508/01]), the Scottish Funding Council, Heriot-Watt University and the University of Edinburgh.

G. dos Reis acknowledges support from the *Fundação para a Ciência e a Tecnologia* (Portuguese Foundation for Science and Technology) through the project [UID/MAT/00297/2013] (Centro de Matemática e Aplicações CMA/FCT/UNL).

A Proofs

A.1 Proof of Lemma 2.8

We now provide the proof of Lemma 2.8, all terms used have the same definition as they did when the Lemma was stated. Throughout we assume $i \neq h$, thus from Assumption 2.7 $\mathbb{P}_{\mathbf{Q}}(X(t) = j | X(0) = i) > 0$ for all $j \in \{1, \dots, h\}$ and $t > 0$. The first inequality we prove is the lower bound on the expected number of jumps. Following the assumptions in Lemma 2.8 and time homogeneity we make the observation

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(T) | \mathbf{P}] \geq P_{ij}^u \mathbb{P}_{\mathbf{Q}}(K_{ij}(t) \geq 1 | X(0) = i, X(t) = j).$$

The above inequality holds because we are only considering $X(0) = i$, $X(t) = j$ and not all possible combinations of start and end states, moreover, $\mathbb{P}_{\mathbf{Q}}(K_{ij} \geq 1 | X(0) = i, X(t) = j) \leq \sum_{n=1}^{\infty} n \mathbb{P}_{\mathbf{Q}}(K_{ij} = n | X(0) = i, X(t) = j)$. We further observe,

$$\mathbb{P}_{\mathbf{Q}}(K_{ij} \geq 1 | X(0) = i, X(t) = j) \geq \frac{q_{ij}}{-q_{ii}}.$$

Thus the lower bound in inequality (2.12) can be easily obtained. We now prove the upper bound on the expected number of jumps. The first observation we make is for all $\nu \in \{1, \dots, h\}$,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(T) | X(0) = i, X(t) = \nu] = \sup_{\mu \in \{1, \dots, h\}} \mathbb{E}_{\mathbf{Q}}[K_{ij}(T) | X(0) = \mu, X(t) = \nu].$$

To see this, let $\mu \neq i$, then denote by τ_i the first time the process enters state i (if $\mathbb{P}_{\mathbf{Q}}(X(t) = i | X(0) = \mu) = 0$ for $t > 0$, then the result is trivial), by the law of total probability we find,

$$\begin{aligned} \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = \mu, X(t) = \nu] \\ = \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = \mu, X(t) = \nu, \tau_i < t] \mathbb{P}_{\mathbf{Q}}(\tau_i < t | X(0) = \mu, X(t) = \nu) \\ + \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = \mu, X(t) = \nu, \tau_i \geq t] \mathbb{P}_{\mathbf{Q}}(\tau_i \geq t | X(0) = \mu, X(t) = \nu). \end{aligned}$$

The second term is zero. Then, using the Markov property we obtain,

$$\begin{aligned} \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = \mu, X(t) = \nu] &\leq \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(\tau_i) = i, X(t) = \nu, \tau_i < t] \\ &\leq \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = i, X(t) = \nu]. \end{aligned}$$

Consequently from this observation and (2.11) we obtain,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(T) | \mathbf{P}] \leq hN \sum_{\nu=1}^h \mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = i, X(t) = \nu].$$

Observe that,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = i, X(t) = \nu] = \frac{\mathbb{E}_{\mathbf{Q}}[K_{ij}(t) \mathbb{1}_{\{X(t)=\nu\}} | X(0) = i]}{\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = i)} \leq \frac{\mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = i]}{\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = i)}.$$

The numerator is easy to bound by considering the expected number of jumps out of i ,

$$\mathbb{E}_{\mathbf{Q}}[K_{ij}(t) | X(0) = i] \leq -q_{ii}t.$$

The denominator requires further analysis, firstly, let $n = |i - \nu|$, and therefore by Assumption 2.7 we can go from state i to ν in n jumps, w.l.o.g. let $i \geq \nu$ (it will be come clear that the ordering does not matter). Firstly, if $i = \nu$ then,

$$\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = i) \geq e^{q_{ii}t}.$$

For $i > \nu$, we use the Markov property to obtain,

$$\mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = i) \geq \prod_{a=1}^n \mathbb{P}_{\mathbf{Q}}\left(X\left(\frac{a}{n}t\right) = i + a \mid X\left(\frac{a-1}{n}t\right) = i + a - 1\right).$$

Conditioning on X only making one jump in each increment we obtain,

$$\begin{aligned} \mathbb{P}_{\mathbf{Q}}(X(t) = \nu | X(0) = i) &\geq \prod_{a=1}^n \frac{q_{i+a-1, i+a}}{-q_{i+a-1, i+a-1}} (-q_{i+a-1, i+a-1}) t \exp\{q_{i+a-1, i+a-1}t\} \\ &\geq \prod_{a=1}^n \epsilon t \exp\{-ht/\epsilon\}. \end{aligned}$$

As $n \leq h$ and the terms are strictly smaller than 1, the sought result follows (independent of $\nu \neq i$).

The last inequality to prove concerns the holding times. By taking for $P_{ii}^u > 0$,

$$\mathbb{E}_{\mathbf{Q}}[S_i(T) | \mathbf{P}] \geq P_{ii}^u \mathbb{E}_{\mathbf{Q}}[S_i(t) | X(0) = i, X(t) = i] \geq P_{ii}^u t \exp\{q_{ii}t\},$$

where the final inequality follows by simply considering the case of no jumps. We can then apply the bounds from Assumption 2.7 to complete the inequality.

A.2 Proof of Theorem 2.13

We recall from [Wil67], [TC03] that for a square matrix \mathbf{M} whose elements depend on a vector of parameters $\{\lambda_1, \dots, \lambda_r\}$ (for $r \in \mathbb{N}$), the following identity holds

$$\frac{\partial e^{\mathbf{M}(\lambda)t}}{\partial \lambda_i} = \int_0^t e^{(t-u)\mathbf{M}(\lambda)} \frac{\partial \mathbf{M}(\lambda)}{\partial \lambda_i} e^{u\mathbf{M}(\lambda)} du, \quad (\text{A.1})$$

for all $i \in \{1, \dots, r\}$. Let $\mu, \nu, \alpha, \beta \in \{1, \dots, h\}$. Recalling Proposition 2.4, differentiating $\mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y]$ w.r.t. $q_{\alpha\beta}$ yields,

$$\begin{aligned} \frac{\partial}{\partial q_{\alpha\beta}} \mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y] &= \sum_{s=1}^{n-1} - (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(\frac{\partial}{\partial q_{\alpha\beta}} e^{\mathbf{Q}(t_{s+1}-t_s)} \right)_{y_s, y_{s+1}} (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \\ &\quad + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(\frac{\partial}{\partial q_{\alpha\beta}} e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}}. \end{aligned}$$

Note that although the expected value of K only depends on individual elements of the matrix and not the full matrix, we are still able to use the differentiation result since $A_{ij} = \mathbf{e}_i^\top \mathbf{A} \mathbf{e}_j$. Hence, from (A.1) we obtain,

$$\begin{aligned} \frac{\partial}{\partial q_{\alpha\beta}} \mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y] &= \sum_{s=1}^{n-1} - (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(\int_0^t e^{(t-u)\mathbf{Q}} \frac{\partial \mathbf{Q}}{\partial q_{\alpha\beta}} e^{u\mathbf{Q}} du \right)_{y_s, y_{s+1}} (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \\ &\quad + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(\int_0^t e^{(t-u)\mathbf{C}_\gamma^{(\mu\nu)}} \frac{\partial \mathbf{C}_\gamma^{(\mu\nu)}}{\partial q_{\alpha\beta}} e^{u\mathbf{C}_\gamma^{(\mu\nu)}} du \right)_{y_s, h+y_{s+1}}. \end{aligned}$$

Clearly, since $q_{\alpha\beta}$ appears twice in \mathbf{Q} ,

$$\frac{\partial \mathbf{Q}}{\partial q_{\alpha\beta}} = \mathbf{e}_\alpha \mathbf{e}_\beta^\top - \mathbf{e}_\alpha \mathbf{e}_\alpha^\top, \quad \text{and} \quad \frac{\partial \mathbf{C}_\gamma^{(\mu\nu)}}{\partial q_{\alpha\beta}} = \begin{bmatrix} \mathbf{e}_\alpha \mathbf{e}_\beta^\top - \mathbf{e}_\alpha \mathbf{e}_\alpha^\top & \mathbf{e}_\mu \mathbf{e}_\nu^\top \delta_{\mu\alpha} \delta_{\nu\beta} \\ 0 & \mathbf{e}_\alpha \mathbf{e}_\beta^\top - \mathbf{e}_\alpha \mathbf{e}_\alpha^\top \end{bmatrix}.$$

Then, by [VL78] we can solve these integrals explicitly to obtain,

$$\begin{aligned} \frac{\partial}{\partial q_{\alpha\beta}} \mathbb{E}_{\mathbf{Q}}[K_{\mu\nu}(t)|y] &= \sum_{s=1}^{n-1} - (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} (e^{\mathbf{C}_\gamma^{(\mu\nu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \\ &\quad + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(e^{\mathbf{C}_\psi^{(\alpha\beta, \mu\nu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}}, \end{aligned}$$

again $\mathbf{C}_\eta^{(\alpha\beta)}$ and $\mathbf{C}_\psi^{(\alpha\beta, \mu\nu)}$ are as defined in the Theorem's statement.

Therefore, we have a closed form expression for the derivative of expected jumps w.r.t. $q_{\alpha\beta}$. Applying a similar argument for the expected holding time we obtain,

$$\begin{aligned} \frac{\partial}{\partial q_{\alpha\beta}} \mathbb{E}_{\mathbf{Q}}[S_\mu(t)|y] &= \sum_{s=1}^{n-1} - (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-2} \left(e^{\mathbf{C}_\eta^{(\alpha\beta)}(t_{s+1}-t_s)} \right)_{y_s, h+y_{s+1}} (e^{\mathbf{C}_\phi^{(\mu)}(t_{s+1}-t_s)})_{y_s, h+y_{s+1}} \\ &\quad + (e^{\mathbf{Q}(t_{s+1}-t_s)})_{y_s, y_{s+1}}^{-1} \left(e^{\mathbf{C}_\omega^{(\alpha\beta, \mu)}(t_{s+1}-t_s)} \right)_{y_s, 3h+y_{s+1}}, \end{aligned}$$

where $\mathbf{C}_\omega^{(\alpha\beta, \mu)}$ is as defined in the Theorem. Combining these yields the required result.

B Overview of Markov Chain Monte Carlo algorithm

Markov Chain Monte Carlo (MCMC) has been highly successful in other fields. For details on the theory the reader can consult texts such as [GRS96]. Algorithms for implementing MCMC to estimate a generator, from discrete observations are discussed in [BS05] and [BS09].

MCMC differs from the EM in the sense that EM estimates the set of parameters which maximises the likelihood function, while MCMC samples from the posterior distribution. Namely, given some data D , the posterior distribution of parameters θ is $\pi(\theta|D)$, which by Bayes' theorem is,

$$\pi(\theta|D) = \frac{\pi(D|\theta)\pi(\theta)}{\int \pi(D|\theta)\pi(\theta)d\theta},$$

with $\pi(D|\theta)$ denoting the likelihood and $\pi(\theta)$ the prior distribution. MCMC obtains the best guess of θ by sampling from $\pi(\theta|D)$ and taking the Monte Carlo approximation of the expected value. The reason the expectation is our best guess is due to the fact we use both the data (likelihood) but also our experience on what the outcome should approximately be (the prior). Although the prior can be extremely useful in stopping 'bad' answers it is also a criticism of MCMC due to so-called prior sensitivity.

Remark B.1. Here we purely discuss MCMC to sample from the posterior, algorithms which approximate the maximum likelihood in the presence of missing data do exist, but are more useful when for example one cannot explicitly write the E step in the EM algorithm (see [GC93]).

Similar to the case of the EM algorithm the problem faced here is missing data. Namely we wish to consider the so-called posterior distribution of the generator matrix \mathbf{Q} , which we denote by $\pi(\mathbf{Q}|D)$ (although it is common to suppress the data and only write $\pi(\mathbf{Q})$). The difficulty is, in its current state this is an extremely hard distribution to evaluate so we augment with an auxiliary variable X (see [GRS96, p.105] and [BG93]). In general X need not require an interpretation, although here it will correspond to the full Markov chain. In order to generate realisations of $\pi(\mathbf{Q}|D)$, we specify the conditional distribution $\pi(X|\mathbf{Q}, D)$ which provides the joint distribution $\pi(\mathbf{Q}, X|D) = \pi(\mathbf{Q}|D)\pi(X|\mathbf{Q}, D)$ and therefore the marginal distribution of \mathbf{Q} is $\pi(\mathbf{Q}|D)$. One can then sample from the marginal distribution by using any sampling method that preserves the joint distribution $\pi(\mathbf{Q}, X|D)$ (and by extension $\pi(\mathbf{Q}|D)$), such as Gibbs or Metropolis Hastings.

The method used in [BS05] and [BS09] is the data augmentation algorithm from [TW87] (see also [LR02, p.200]). We specify the prior distribution $\pi(\mathbf{Q})$ and take a realisation from this distribution, $\mathbf{Q}^{(0)}$, we then construct a sequence $\{\mathbf{Q}^{(k)}, X^{(k)}\}$, for $k = 1, \dots, M$ by:

1. Draw, $X^{(k)} \sim \pi(X|\mathbf{Q}^{(k-1)}, D)$.
2. Draw, $\mathbf{Q}^{(k)} \sim \pi(\mathbf{Q}|X^{(k)}, D) = \pi(\mathbf{Q}|X^{(k)})$ (since $X^{(k)}$ is richer than D).
3. Save $\{\mathbf{Q}^{(k)}, X^{(k)}\}$ and take $k = k + 1$.

Under mild conditions (see [GRS96, Chapter 4]), after some burn-in n , the sequence $\{\mathbf{Q}^{(k)}, X^{(k)}\}$ for $k \geq n$ has the same distribution as $\pi(\mathbf{Q}, X|D)$. Moreover, the marginals also have the correct distribution, namely, $\{\mathbf{Q}^{(k)}\} \sim \pi(\mathbf{Q}|D)$ for $k \geq n$. Therefore we estimate the generator matrix by, $\frac{1}{M-n+1} \sum_{k=n}^M \mathbf{Q}^{(k)}$.

For the choice of prior, $\pi(\mathbf{Q})$, [BS05] suggest a prior from the gamma distribution with shape α_{ij} and scale $1/\beta_i$. Hence, $q_{ij} \sim \Gamma(\alpha_{ij}, 1/\beta_i)$, where $\alpha_{ij}, \beta_i \geq 0, \forall i \neq j \in \{1, \dots, h\}$. With this choice, the prior is a conjugate prior. Although this prior has some drawbacks, we note, by assuming the prior to follow a Gamma distribution we effectively bound the parameter space, therefore there is no need to make the space compact. Noting that the posterior distribution of X is equivalent to the likelihood i.e. $\pi(X|\mathbf{Q}) = L_t(X; \mathbf{Q})$, one has

$$\pi(\mathbf{Q}|X, D) = \pi(\mathbf{Q}|X) = \frac{\pi(\mathbf{Q}, X)}{\pi(X)} \propto L_t(X; \mathbf{Q})\pi(\mathbf{Q}).$$

From the likelihood of a CTMC and the assumption on the prior we infer that,

$$L_t(X; \mathbf{Q})\pi(\mathbf{Q}) \propto \prod_{i=1}^h \prod_{j \neq i} q_{ij}^{K_{ij}(t)} e^{-S_i(t)q_{ij}} \prod_{i=1}^h \prod_{j \neq i} q_{ij}^{\alpha_{ij}-1} e^{-\beta_i q_{ij}} = \prod_{i=1}^h \prod_{j \neq i} q_{ij}^{K_{ij}(t)+\alpha_{ij}-1} e^{-(S_i(t)+\beta_i)q_{ij}}.$$

We do not have equality here since there is no normalisation term. We generate q_{ij} with $i \neq j$ from the distribution $\Gamma(K_{ij}(t) + \alpha_{ij}, 1/(S_i(t) + \beta_i))$ (since each q_{ij} is independent).

References

- [BDK⁺02] A. Bangia, F. X. Diebold, A. Kronimus, C. Schagen, and T. Schuermann, *Ratings migration and the business cycle, with application to credit portfolio stress testing*, Journal of Banking & Finance **26** (2002), no. 2, 445–474.
- [BG93] J. Besag and P. J. Green, *Spatial statistics and Bayesian computation*, Journal of the Royal Statistical Society. Series B (Methodological) (1993), 25–37.
- [BMNS02] M. Bladt, B. Meini, M. F. Neuts, and B. Sericola, *Distributions of reward functions on continuous-time Markov chains*, Matrix-analytic methods (2002), 39–62.
- [BMS14] D. Brigo, J.-F. Mai, and M. A. Scherer, *Consistent iterated simulation of multi-variate default times: a Markovian indicators characterization*, Available at SSRN 2274369 (2014).
- [BS05] M. Bladt and M. Sorensen, *Statistical inference for discretely observed Markov jump processes*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **67** (2005), no. 3, 395–410.
- [BS09] M. Bladt and M. Sorensen, *Efficient estimation of transition rates between credit ratings from observations at discrete time points*, Quantitative Finance **9** (2009), no. 2, 147–160.
- [Can04] R. Cantor, *An introduction to recent research on credit ratings*, Journal of Banking & Finance **28** (2004), no. 11, 2565–2573.
- [CDS10] R. Cont, R. Deguest, and G. Scandolo, *Robustness and sensitivity analysis of risk measurement procedures*, Quantitative Finance **10** (2010), no. 6, 593–606.
- [CHL04] J. H. E. Christensen, E. Hansen, and D. Lando, *Confidence sets for continuous-time rating transition probabilities*, Journal of Banking & Finance **28** (2004), no. 11, 2575–2602.
- [Cut73] J. R. Cuthbert, *The logarithm function for finite-state Markov semi-groups*, Journal of the London Mathematical Society **2** (1973), no. 3, 524–532.
- [DY07] D. Dehay and J.-F. Yao, *On likelihood estimation for discretely observed Markov jump processes*, Australian & New Zealand Journal of Statistics **49** (2007), no. 1, 93–107.
- [Fer14] J.-D. Fermanian, *The limits of granularity adjustments*, Journal of Banking & Finance **45** (2014), 9–25.
- [FS08] H. Frydman and T. Schuermann, *Credit rating dynamics and Markov mixture models*, Journal of Banking & Finance **32** (2008), no. 6, 1062–1075.
- [GC93] A. E. Gelfand and B. P. Carlin, *Maximum-likelihood estimation for constrained-or missing-data models*, Canadian Journal of Statistics **21** (1993), no. 3, 303–311.
- [GFB97] G. M. Gupton, C. C. Finger, and M. Bhatia, *Creditmetrics: technical document*, JP Morgan & Co., 1997.
- [GRS96] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, *Introducing Markov chain Monte Carlo*, Markov chain Monte Carlo in practice **1** (1996), 19.
- [Ina06] Y. Inamura, *Estimating continuous time transition matrices from discretely observed data*, Citeseer, 2006. (No. 06-E-7). Bank of Japan.
- [IRW01] R. B. Israel, J. S. Rosenthal, and J. Z. Wei, *Finding generators for Markov chains via empirical transition matrices, with applications to credit ratings*, Mathematical finance **11** (2001), no. 2, 245–265.
- [JLT97] R. A. Jarrow, D. Lando, and S. M. Turnbull, *A Markov model for the term structure of credit risk spreads*, Review of Financial studies **10** (1997), no. 2, 481–523.
- [Kor12] M. W. Korolkiewicz, *A dependent hidden Markov model of credit quality*, International Journal of Stochastic Analysis **2012** (2012).
- [KS01] A. Kreinin and M. Sidelnikova, *Regularization algorithms for transition matrices*, Algo Research Quarterly **4** (2001), no. 1/2, 23–40.
- [KS97] U. Küchler and M. Sorensen, *Exponential families of stochastic processes*, Vol. 3, Springer Science & Business Media, 1997.
- [KW13] A. Kremer and R. Weißbach, *Consistent estimation for discretely observed Markov jump processes with an absorbing state*, Statistical Papers **54** (2013), no. 4, 993–1007.
- [KW14] A. Kremer and R. Weißbach, *Asymptotic normality for discretely observed Markov jump processes with an absorbing state*, Statistics & Probability Letters **90** (2014), 136–139.
- [Lin11] L. Lin, *Roots of stochastic matrices and fractional matrix powers*, Ph.D. Thesis, 2011.
- [LKN⁺11] K. Long, S. C. Keenan, R. Neagu, J. A. Ellis, and J. W. Black, *The computation of optimised credit transition matrices*, Journal of Risk Management in Financial Institutions **4** (2011), no. 4, 370–391.

- [LR02] R. J. A. Little and D. B. Rubin, *Statistical analysis with missing data*, John Wiley & Sons, 2002.
- [LS02] D. Lando and T. M. Skodeberg, *Analyzing rating transitions and rating drift with continuous observations*, Journal of Banking and Finance **26** (2002), no. 2, 423–444.
- [MK07] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*, Vol. 382, John Wiley & Sons, 2007.
- [Nor98] J. R. Norris, *Markov chains*, Cambridge university press, 1998.
- [Oak99] D. Oakes, *Direct calculation of the information matrix via the EM*, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **61** (1999), no. 2, 479–482.
- [RT15] M. Rutkowski and S. Tarca, *Regulatory capital modeling for credit risk*, International Journal of Theoretical and Applied Finance **18** (2015), no. 05, 1550034.
- [Sup03] B. C. o. B. Supervision, *The new Basel capital accord* (2003).
- [Sup13] B. C. o. B. Supervision, *Fundamental review of the trading book: A revised market risk framework* (2013).
- [TC03] H. Tsai and K. Chan, *A note on parameter differentiation of matrix exponentials, with applications to continuous-time modelling*, Bernoulli (2003), 895–919.
- [TÖ04] S. Trück and E. Özturkmen, *Estimation, adjustment and application of transition matrices in credit risk models*, Handbook of computational and numerical methods in finance, 2004, pp. 373–402.
- [TW87] M. A. Tanner and W. H. Wong, *The calculation of posterior distributions by data augmentation*, Journal of the American statistical Association **82** (1987), no. 398, 528–540.
- [VL78] C. Van Loan, *Computing integrals involving the matrix exponential*, Automatic Control, IEEE Transactions on **23** (1978), no. 3, 395–404.
- [Wil67] R. Wilcox, *Exponential operators and parameter differentiation in quantum physics*, Journal of Mathematical Physics **8** (1967), no. 4, 962–982.
- [Wu83] C. F. J. Wu, *On the convergence properties of the EM algorithm*, The Annals of statistics (1983), 95–103.
- [YWZC14] T. Yavin, E. Wang, H. Zhang, and M. A. Clayton, *Transition probability matrix methodology for incremental risk charge*, Journal of Financial Engineering **1** (2014), no. 01.